

# Advanced Tutorial on Implementing GPFgrowth Algorithm

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

## Step 1: Import the GPFgrowth algorithm and pandas data frame

```
In [1]: from PAMI.partialPeriodicFrequentPattern.basic import GPFgrowth as alg
import pandas as pd
```

## Step 2: Specify the following input parameters

```
In [2]: inputFile = 'temporal_T10I4D100K.csv'
separator='¥t'
maxmunPeriodCount=5000
minimumSupportCountList = [100, 150, 200, 250, 300]
#minimumSupport can also specified between 0 to 1. E.g., minSupList = [0.005, 0.006,
minPRCount=0.5
result = pd.DataFrame(columns=['algorithm', 'minSup', 'maxPer', 'minPR', 'patterns', '
#initialize a data frame to store the results of GPFgrowth algorithm
```

## Step 3: Execute the GPFgrowth algorithm using a for loop

```
In [3]: algorithm = 'GPFgrowth' #specify the algorithm name
for minSupCount in minimumSupportCountList:
    obj = alg.GPFgrowth('temporal_T10I4D100K.csv', minSup=minSupCount, maxPer=maxmunf
    obj.startMine()
    #store the results in the data frame
    result.loc[result.shape[0]] = [algorithm, minSupCount, maxmunPeriodCount, minPRCou
    print(result)
```

	algorithm	minSup	maxPer	minPR	patterns	runtime	memory
0	GPFgrowth	100	5000	0.5	27532	59.388855	744574976
	algorithm	minSup	maxPer	minPR	patterns	runtime	memory
0	GPFgrowth	100	5000	0.5	27532	59.388855	744574976
1	GPFgrowth	150	5000	0.5	19178	63.220936	904560640
	algorithm	minSup	maxPer	minPR	patterns	runtime	memory
0	GPFgrowth	100	5000	0.5	27532	59.388855	744574976
1	GPFgrowth	150	5000	0.5	19178	63.220936	904560640
2	GPFgrowth	200	5000	0.5	13337	69.218870	1063096320
	algorithm	minSup	maxPer	minPR	patterns	runtime	memory
0	GPFgrowth	100	5000	0.5	27532	59.388855	744574976
1	GPFgrowth	150	5000	0.5	19178	63.220936	904560640
2	GPFgrowth	200	5000	0.5	13337	69.218870	1063096320
3	GPFgrowth	250	5000	0.5	7810	70.672142	1224404992
	algorithm	minSup	maxPer	minPR	patterns	runtime	memory
0	GPFgrowth	100	5000	0.5	27532	59.388855	744574976
1	GPFgrowth	150	5000	0.5	19178	63.220936	904560640
2	GPFgrowth	200	5000	0.5	13337	69.218870	1063096320
3	GPFgrowth	250	5000	0.5	7810	70.672142	1224404992
4	GPFgrowth	300	5000	0.5	4687	74.671376	1403801600

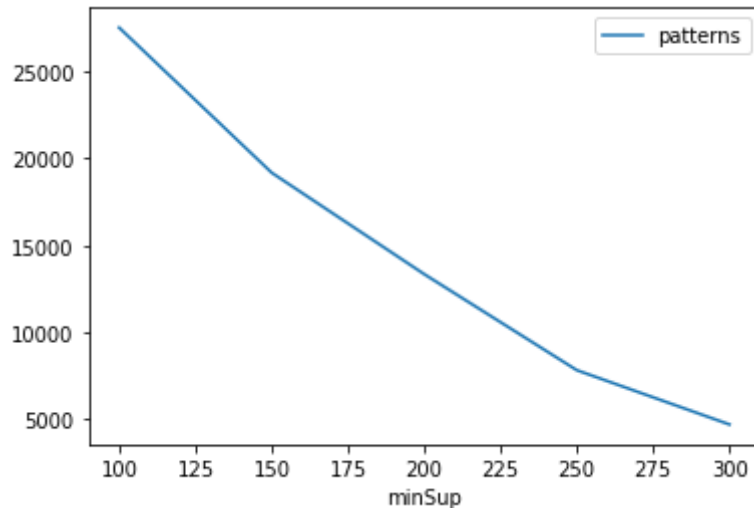
## Step 5: Visualizing the results

### Step 5.1 Importing the plot library

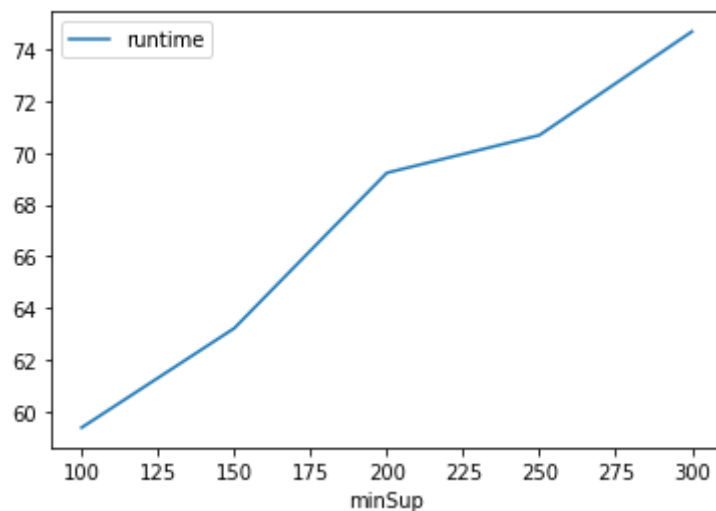
```
In [4]: from PAMI.extras.graph import plotLineGraphsFromDataFrame as plt
```

### Step 5.2. Plotting the number of patterns

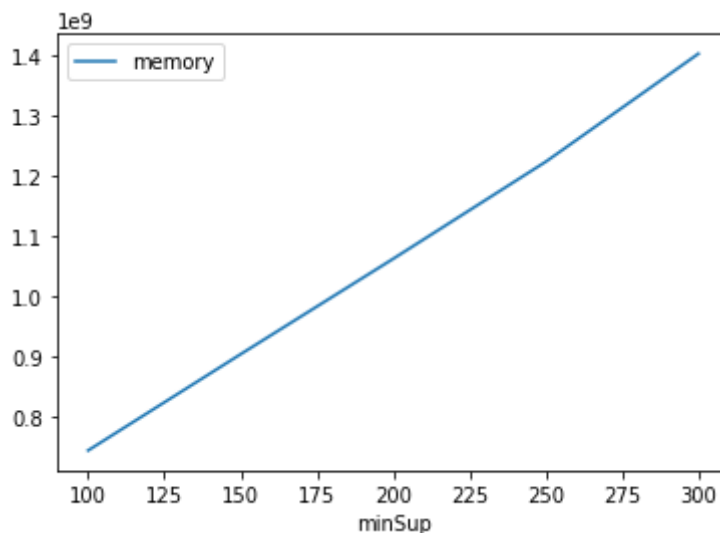
```
In [5]: ab = plt.plotGraphsFromDataFrame(result)
ab.plotGraphsFromDataFrame() #drawPlots()
```



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

Latex files generated successfully