COL761 Assignment-2

Manshi Sagar - 2020CS50429

Rishita Agrawal - 2020CS50439

Rajat Bhardwaj - 2020CS50436

Abstract

In the first part of this assignment, we compare the performance of frequent subgraph mining algorithms, namely, gSpan, FSG (also known as PAFI), and Gaston. In the second part of the assignment, given a graph database, we identify the subgraphs crucial for classification and use them to convert each database graph into a binary presence/absence feature vector X consisting of at most 100 dimensions.

1 Question 1

1.1 gSpan

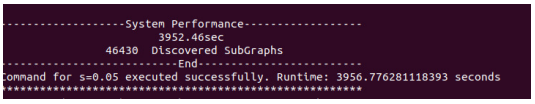


Figure 1: gSpan output at 5% support

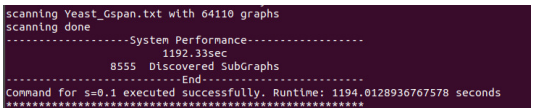


Figure 2: gSpan output at 10% support

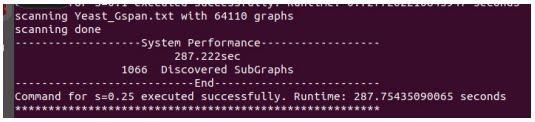


Figure 3: gSpan output at 25% support

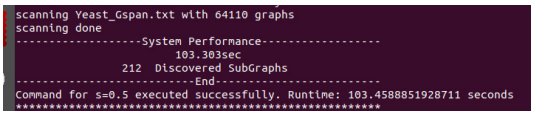


Figure 4: gSpan output at 50% support

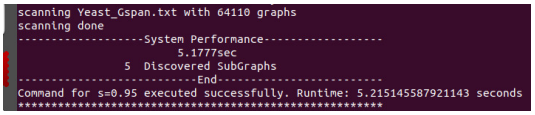


Figure 5: gSpan output at 95% support

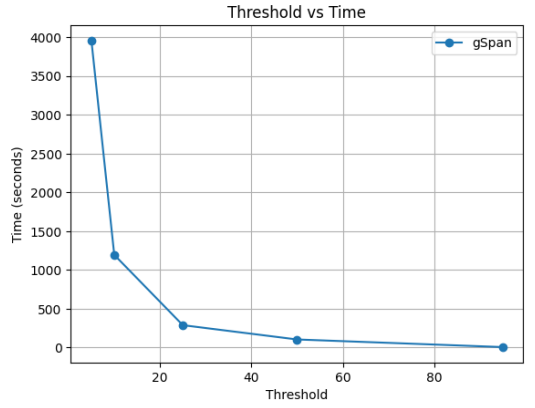


Figure 6: Plot of gSpan

Table 1: gSpan statistics

Data		
Support %	Frequent subgraphs	Execution time
5	46430	3956.77[sec]
10	8555	1194.01[sec]
25	1066	287.75[sec]
50	212	103.45[sec]
95	5	5.177[sec]

1.2 FSG (or PAFI)

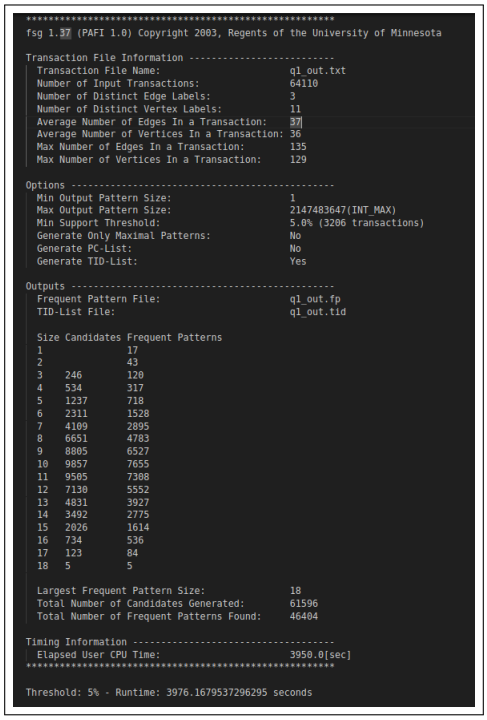


Figure 7: FSG output at 5% support

Table 2: FSG statistics

Data		
Support %	Frequent subgraphs	Execution time
5	46404	3950.0[sec]
10	8549	1294.3[sec]
25	1062	351.9[sec]
59	208	108.0[sec]
95	3	2.2[sec]

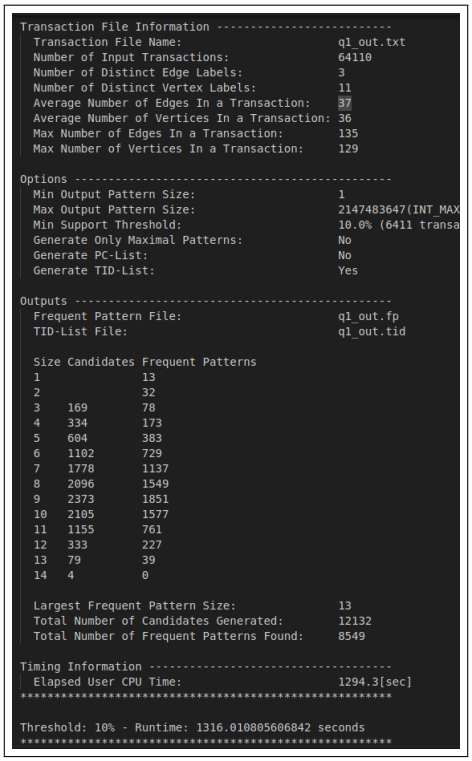


Figure 8: FSG output at 10% support

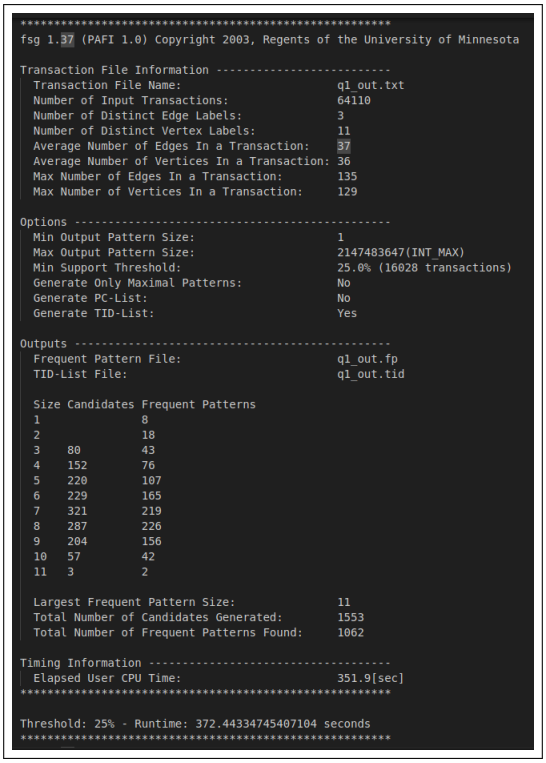
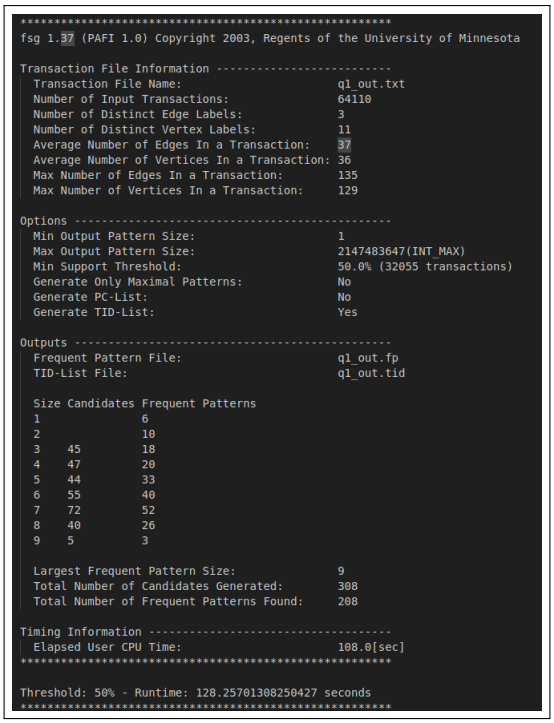
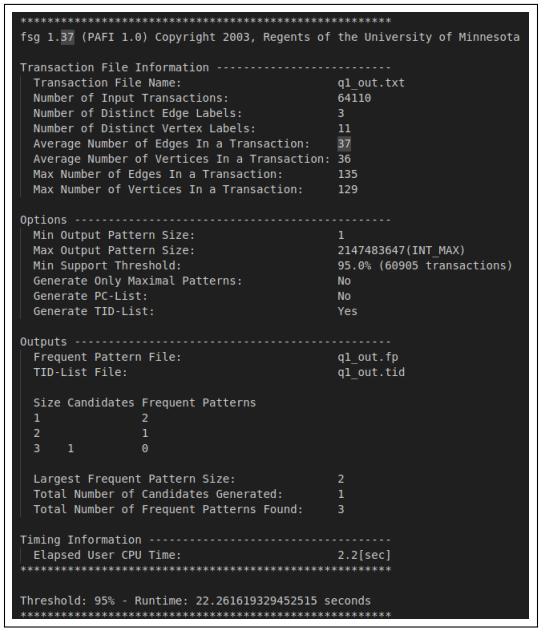


Figure 9: FSG output at 25% support



```
*******************************************************
fsg 1.37 (PAFI 1.0) Copyright 2003, Regents of the University of Minnesota

Transaction File Information -------------------------
  Transaction File Name:                        q1_out.txt
  Number of Input Transactions:                 64110
  Number of Distinct Edge Labels:               3
  Number of Distinct Vertex Labels:             11
  Average Number of Edges In a Transaction:     37
  Average Number of Vertices In a Transaction: 36
  Max Number of Edges In a Transaction:         135
  Max Number of Vertices In a Transaction:      129

Options ----------------------------------------------
  Min Output Pattern Size:                      1
  Max Output Pattern Size:                      2147483647(INT_MAX)
  Min Support Threshold:                        50.0% (32055 transactions)
  Generate Only Maximal Patterns:               No
  Generate PC-List:                             No
  Generate TID-List:                            Yes

Outputs ----------------------------------------------
  Frequent Pattern File:                        q1_out.fp
  TID-List File:                                q1_out.tid

  Size Candidates Frequent Patterns
  1              6
  2              10
  3    45        18
  4    47        20
  5    44        33
  6    55        40
  7    72        52
  8    40        26
  9    5         3

  Largest Frequent Pattern Size:                9
  Total Number of Candidates Generated:         308
  Total Number of Frequent Patterns Found:      208

Timing Information -----------------------------------
  Elapsed User CPU Time:                        108.0[sec]
*******************************************************

Threshold: 50% - Runtime: 128.25701308250427 seconds
*******************************************************
```

Figure 10: FSG output at 50% support



```
*******************************************************
fsg 1.37 (PAFI 1.0) Copyright 2003, Regents of the University of Minnesota

Transaction File Information -------------------------
  Transaction File Name:                        q1_out.txt
  Number of Input Transactions:                 64110
  Number of Distinct Edge Labels:               3
  Number of Distinct Vertex Labels:             11
  Average Number of Edges In a Transaction:     37
  Average Number of Vertices In a Transaction: 36
  Max Number of Edges In a Transaction:         135
  Max Number of Vertices In a Transaction:      129

Options ----------------------------------------------
  Min Output Pattern Size:                      1
  Max Output Pattern Size:                      2147483647(INT_MAX)
  Min Support Threshold:                        95.0% (60905 transactions)
  Generate Only Maximal Patterns:               No
  Generate PC-List:                             No
  Generate TID-List:                            Yes

Outputs ----------------------------------------------
  Frequent Pattern File:                        q1_out.fp
  TID-List File:                                q1_out.tid

  Size Candidates Frequent Patterns
  1              2
  2              1
  3    1         0

  Largest Frequent Pattern Size:                2
  Total Number of Candidates Generated:         1
  Total Number of Frequent Patterns Found:      3

Timing Information -----------------------------------
  Elapsed User CPU Time:                        2.2[sec]
*******************************************************

Threshold: 95% - Runtime: 22.261619329452515 seconds
*******************************************************
```

Figure 11: FSG output at 95% support

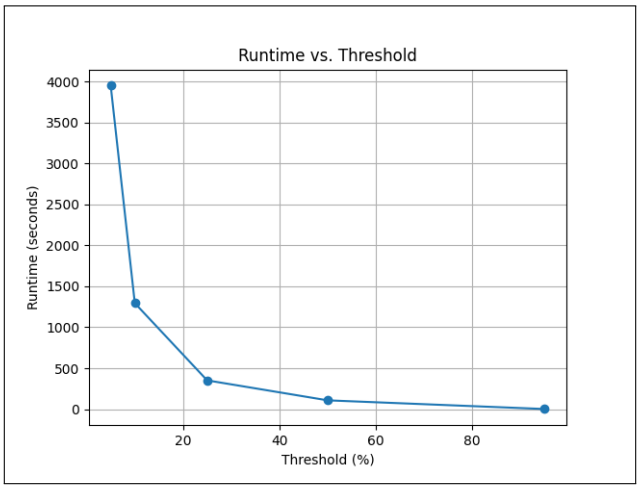


Figure 12: Plot of FSG

1.3 Gaston

```
GASTON GrAph, Sequences and Tree ExtractiON algorithm
Version 1.0 with Occurrence Lists
Siegfried Nijssen, LIACS, 2004
Read
Edgecount
Reorder
Root 0
Root 1
Root 2
Root 3
Root 4
Root 5
Frequent 2 cyclic graphs: 0 real trees: 0 paths: 17 total: 17
Frequent 3 cyclic graphs: 0 real trees: 0 paths: 43 total: 43
Frequent 4 cyclic graphs: 0 real trees: 34 paths: 86 total: 120
Frequent 5 cyclic graphs: 0 real trees: 158 paths: 159 total: 317
Frequent 6 cyclic graphs: 0 real trees: 508 paths: 210 total: 718
Frequent 7 cyclic graphs: 4 real trees: 1278 paths: 246 total: 1528
Frequent 8 cyclic graphs: 8 real trees: 2602 paths: 286 total: 2896
Frequent 9 cyclic graphs: 60 real trees: 4445 paths: 278 total: 4783
Frequent 10 cyclic graphs: 165 real trees: 6151 paths: 212 total: 6528
Frequent 11 cyclic graphs: 334 real trees: 7173 paths: 150 total: 7657
Frequent 12 cyclic graphs: 487 real trees: 6732 paths: 98 total: 7317
Frequent 13 cyclic graphs: 521 real trees: 4992 paths: 41 total: 5554
Frequent 14 cyclic graphs: 378 real trees: 3531 paths: 19 total: 3928
Frequent 15 cyclic graphs: 301 real trees: 2475 paths: 2 total: 2778
Frequent 16 cyclic graphs: 264 real trees: 1351 paths: 0 total: 1615
Frequent 17 cyclic graphs: 125 real trees: 411 paths: 0 total: 536
Frequent 18 cyclic graphs: 27 real trees: 57 paths: 0 total: 84
Frequent 19 cyclic graphs: 1 real trees: 4 paths: 0 total: 5
TOTAL:
Frequent cyclic graphs: 2675 real trees: 41902 paths: 1847 total: 46424
Approximate total runtime: 142.039s
```

Figure 13: Gaston output at 5% support

```
GASTON GrAph, Sequences and Tree ExtractiON algorithm
Version 1.0 with Occurrence Lists
Siegfried Nijssen, LIACS, 2004
Read
Edgecount
Reorder
Root 0
Root 1
Root 2
Root 3
Root 4
Root 5
Frequent 2 cyclic graphs: 0 real trees: 0 paths: 13 total: 13
Frequent 3 cyclic graphs: 0 real trees: 0 paths: 32 total: 32
Frequent 4 cyclic graphs: 0 real trees: 23 paths: 55 total: 78
Frequent 5 cyclic graphs: 0 real trees: 91 paths: 82 total: 173
Frequent 6 cyclic graphs: 0 real trees: 272 paths: 111 total: 383
Frequent 7 cyclic graphs: 1 real trees: 593 paths: 135 total: 729
Frequent 8 cyclic graphs: 5 real trees: 1014 paths: 118 total: 1137
Frequent 9 cyclic graphs: 30 real trees: 1423 paths: 96 total: 1549
Frequent 10 cyclic graphs: 83 real trees: 1699 paths: 69 total: 1851
Frequent 11 cyclic graphs: 137 real trees: 1403 paths: 37 total: 1577
Frequent 12 cyclic graphs: 112 real trees: 636 paths: 14 total: 762
Frequent 13 cyclic graphs: 46 real trees: 173 paths: 8 total: 227
Frequent 14 cyclic graphs: 7 real trees: 30 paths: 2 total: 39
TOTAL:
Frequent cyclic graphs: 421 real trees: 7357 paths: 772 total: 8550
Approximate total runtime: 49.4854s
```

Figure 14: Gaston output at 10% support

Table 3: Gaston statistics

Data		
Support %	Frequent subgraphs	Execution time
5	46424	142.039s
10	8550	49.4854s
25	1062	16.2419s
50	208	9.0041s
95	3	3.07s

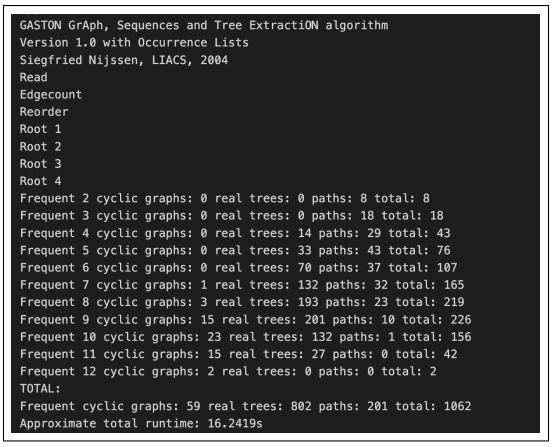


Figure 15: Gaston output at 25% support

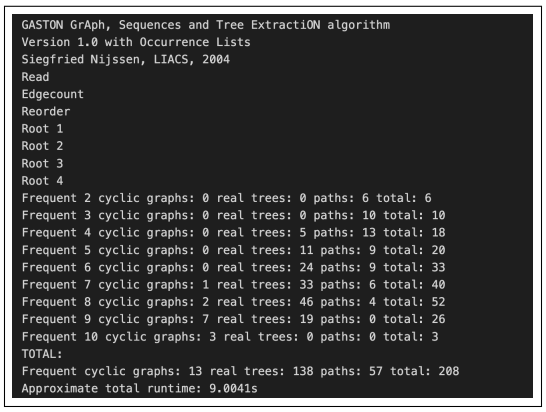


Figure 16: Gaston output at 50% support

Figure 17: Gaston output at 95% support

1.4 Similarities

1. In all algorithms, we see exponential increase in execution time on decreasing support. We can conclude that as the support decreases, more number of frequent subgraphs are mined, and as a result we have to do more subgraph isomorphism tests, so the time increases. 2. We also see that as we increase support percentage, fewer frequent subgraphs are mined, which is expected. The number of subgraphs mined for a lower support are a superset of the subgraphs mined for a higher support.

1.5 General Observations

1. Gaston takes very less time compared to the FSG and gSpan on Yeast dataset. This is probably because its algorithm quickstarts its search by enumerating frequent free trees first.

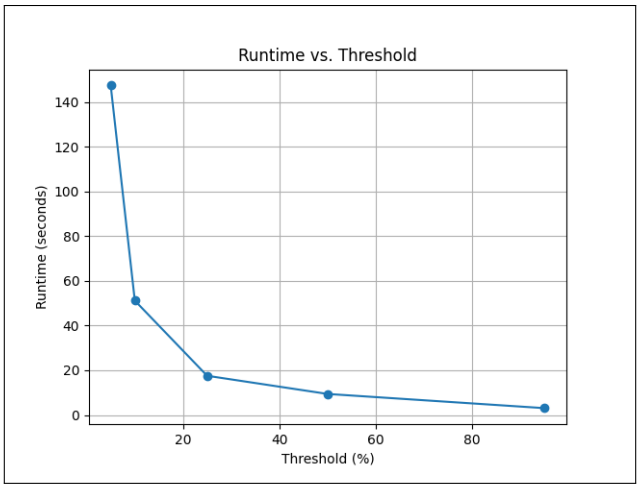


Figure 18: Plot of Gaston

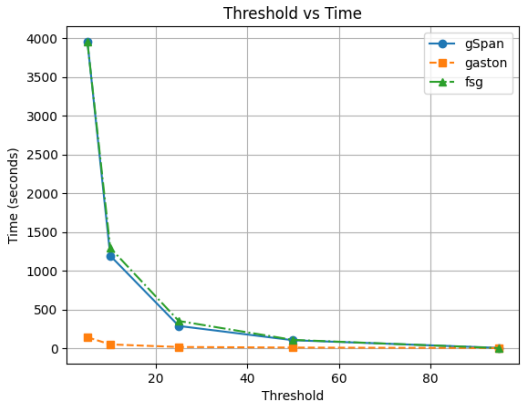


Figure 19: Threshold vs time for Gaston, FSG, gspan

2 Question 2

2.1 Algorithm for mining frequent subgraphs

We used the direct library of FSG (pafi) as given in the assignment pdf for frequent subgraph mining. We chose FSG because it has various flags such as '-t' which gave us the list of graphs in which the subgraph is present. In other algorithms, we would have to do this additional computation using subgraph isomorphism, which would increase our computation time for finding features even if the frequent subgraph mining saved time. Since the datasets were not as big as the Yeast dataset, using FSG minimised the total time for generating features.

2.2 Algorithm for selecting features

We are using the flag '-x' of FSG which generates maximal subgraphs, this reduces the number of subgraphs substantially. Using this flag we ensure that we dont use any 2 such frequent subgraphs are features where one is subgraph of other.
We also tried '-m' and '-M' flag of FSG to keep lower bound and upper bound on the size of frequent subgraphs respectively, but the ROC-AUC decreased on any such changes. We think this is due to the flag '-x'. The flag '-x' works after all the frequent subgraphs are mined but '-m' and '-M' works while mining. After generating all the frequent subgraphs, the '-x' flag removes all smaller subgraphs of the chosen maximal subgraph. The '-M' flag prevents the maximal subgraphs from begin generated and hence less number of smaller graphs are removed, and we get larger number of

frequent subgraphs as output. But we dont want large number of frequent subgraphs, we want the number to be close to 100.
At the end we take top 100 subgraphs from the generated subgraphs are our features.
After fine-tuning on the parameters, we see that optimal results (after classification into 2 classes) were obtained for support = 20 given the time = 30 min. So, we used this value as our final optimal support.

2.3 Observations and Results

Table 4: statistics

Data		
Dataset	ROC-AUC Train	ROC-AUC Test
AIDS	0.9933105468749999	0.9635742187499999
NCI1	0.9140514078520445	0.8396913350027527
Mutagenicity	0.8930139230582966	0.8348402069332302

```
./q2 col761_a2/datasets/AIDS/graphs.txt featureset.txt
******************************************************
fsg 1.37 (PAFI 1.0) Copyright 2003, Regents of the University of Minnesota

Transaction File Information -------------------------
  Transaction File Name:                      temp.txt
  Number of Input Transactions:               2000
  Number of Distinct Edge Labels:             3
  Number of Distinct Vertex Labels:           38
  Average Number of Edges In a Transaction:   16
  Average Number of Vertices In a Transaction: 16
  Max Number of Edges In a Transaction:       103
  Max Number of Vertices In a Transaction:    95

Options ----------------------------------------------
  Min Output Pattern Size:                    1
  Max Output Pattern Size:                    2147483647(INT_MAX)
  Min Support Threshold:                      20.0% (400 transactions)
  Generate Only Maximal Patterns:             Yes
  Generate PC-List:                           Yes
  Generate TID-List:                          Yes

Outputs ----------------------------------------------
  Frequent Pattern File:                      temp.fp
  PC-List File:                               temp.pc
  TID-List File:                              temp.tid

  Size Frequent Patterns
  1    0
  2    2
  3    6
  4    5
  5    3
  6    2

  Largest Frequent Pattern Size:              6
  Total Number of Frequent Patterns Found:    18

Timing Information -----------------------------------
  Elapsed User CPU Time:                      0.2[sec]
******************************************************
Threshold: 20% - Runtime: 0.339 seconds
rishita@rishita-Inspiron-5402:~/col761/A2$ make check
python3 col761_a2/classify.py -g col761_a2/datasets/AIDS/graphs.txt -f featureset.txt
Train ROC_AUC: 0.9933105468749999
Test ROC_AUC: 0.9635742187499999
```

Figure 20: Plot for AIDS

AIDS was giving a ROC-AUC greater than 0.9 in almost all cases (varing support and -m and -M).
NCI1 and Mutagenicity gave contradicting trends. On changing support, if NCI1 was improving,

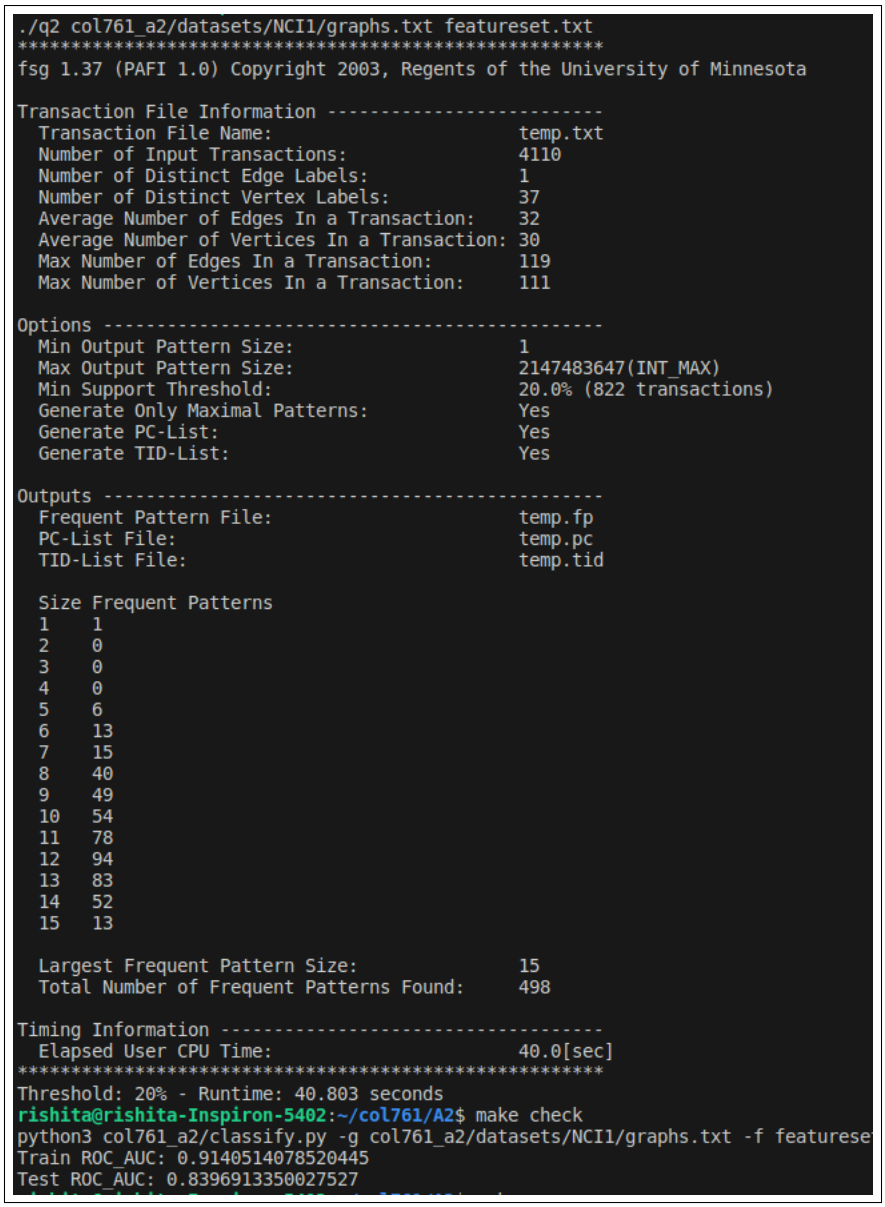


Figure 21: Plot for NCI1

Mutagenicity was degrading in terms of performance and vice versa. So we settled on a value which gave good results (ROC-AUC around 0.8) for both.

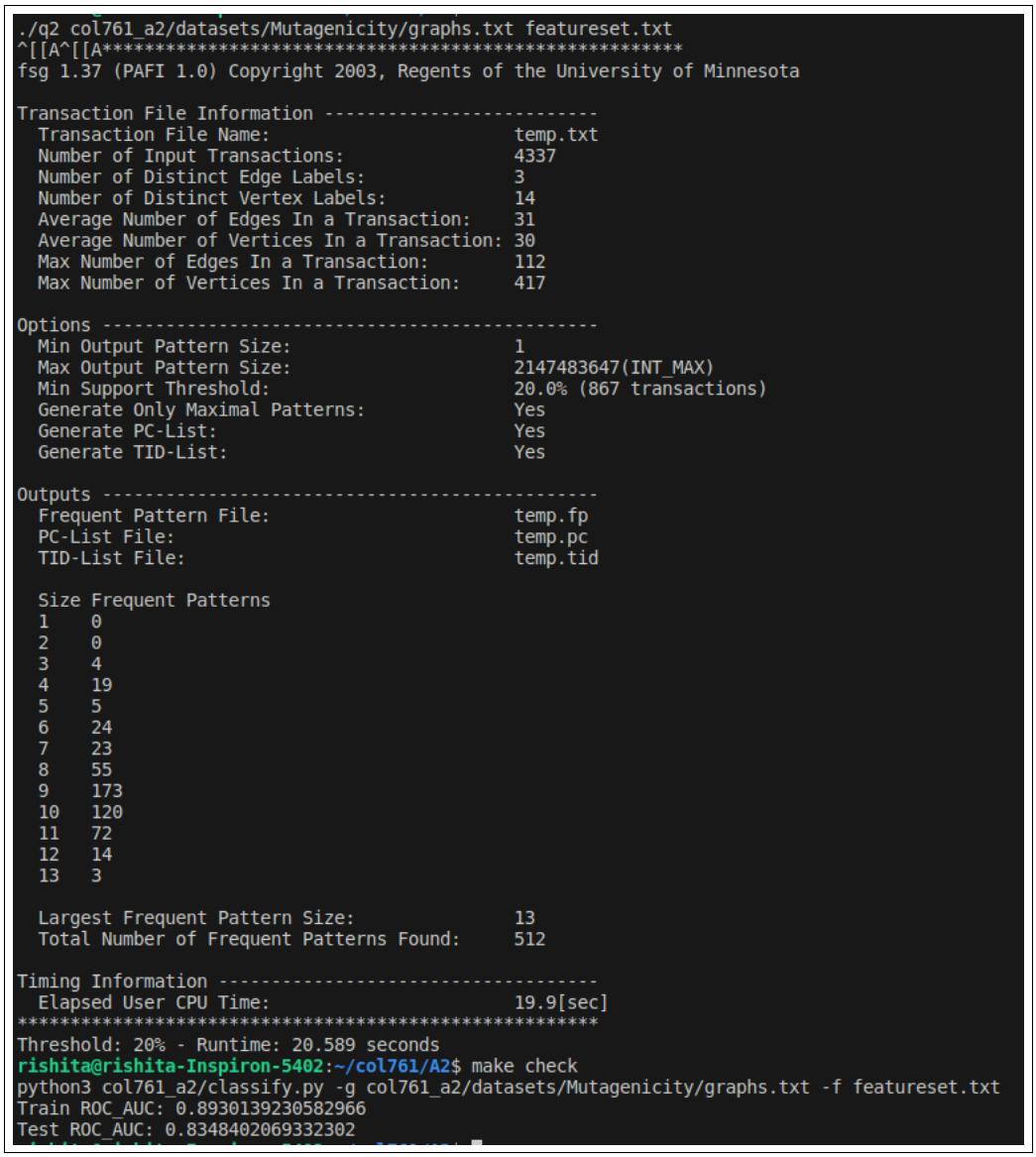


Figure 22: Plot for Mutagenicity

3 Supplementary Material

Authors may wish to optionally include extra information (complete proofs, additional experiments and plots) in the appendix. All such materials should be part of the supplemental material (submitted separately) and should NOT be included in the main submission.

References

[1] gSpan : https://sites.cs.ucsb.edu/ xyan/software/gSpan.htm

[2] FSG : http://glaros.dtc.umn.edu/gkhome/pafi/download

[3] Gaston : https://liacs.leidenuniv.nl/ nijssensgr/gaston/download.html