

Data Mining Final Project
Apriori and FP Growth Algorithms Results

Apriori Algorithm

TEST #1

Give file not of type .arff

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: readme.txt  
The file must be of type .arff
```

TEST #2

Give minimum coverage < 1

```
Bastion-2:cs5001 $ python apriori.py
```

```
Name of the input file: weather.arff  
Minimum coverage (1-n): 0  
Min coverage should be 1 or more, try again:  
Minimum coverage (1-n):
```

TEST #3

Give minimum coverage = 1

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff  
Minimum coverage (1-n): 1  
Maximum size of sets to consider (1-n):
```

TEST #4

Give minimum coverage string value

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff  
Minimum coverage (1-n): string  
Min coverage should be 1 or more, try again:  
Minimum coverage (1-n):
```

TEST #5

Give max size < 0

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): -1
Max size should be 1 or more, try again:
Maximum size of sets to consider (1-n):
```

TEST #6

Give max size = 0

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 0
Max size should be 1 or more, try again:
Maximum size of sets to consider (1-n):
```

TEST #7

Give max size string value

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): string
Max size should be 1 or more, try again:
Maximum size of sets to consider (1-n):
```

TEST #8

Give min accuracy < 0

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): -1
Min accuracy should be between 0-1, try again:
Minimum accuracy (0.0-1.0):
```

TEST #9

Give min accuracy > 1

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 2
Min accuracy should be between 0-1, try again:
Minimum accuracy (0.0-1.0):
```

TEST #10

Give min accuracy a string value

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): string
Min accuracy should be between 0-1, try again:
Minimum accuracy (0.0-1.0):
```

TEST #11

Give report number < 0

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): -1
Report number should be an integer above 0 or 'all', try
again:
Number of rules to report back (1-n, 'all'):
```

TEST #12

Give report number = 0

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): 0
Report number should be an integer above 0 or 'all', try
again:
Number of rules to report back (1-n, 'all'):
```

TEST #13

Give report number a string value other than 'all'

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): string
Report number should be an integer above 0 or 'all', try
again:
Number of rules to report back (1-n, 'all'):
```

TEST #14

Run Apriori on small dataset weather-nominal.arff with nominal values showing all rules

```
r08lnk7x9:CS 5001 Lauren$ python apriori.py
```

```
Name of the input file: weather-nominal.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): all
```

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (4):

```
Item set: ('play', 'yes') Coverage: 9
Item set: ('play', 'no') Coverage: 5
Item set: ('windy', 'TRUE') Coverage: 6
Item set: ('windy', 'FALSE') Coverage: 8
Item set: ('outlook', 'sunny') Coverage: 5
Item set: ('outlook', 'overcast') Coverage: 4
Item set: ('outlook', 'rainy') Coverage: 5
```

Item set: ('temperature', 'hot') Coverage: 4
Item set: ('temperature', 'mild') Coverage: 6
Item set: ('temperature', 'cool') Coverage: 4
Item set: ('humidity', 'high') Coverage: 7
Item set: ('humidity', 'normal') Coverage: 7
Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6
Item set: ('outlook', 'overcast'), ('play', 'yes') Coverage: 4
Item set: ('play', 'yes'), ('temperature', 'mild') Coverage: 4
Item set: ('humidity', 'normal'), ('play', 'yes') Coverage: 6
Item set: ('humidity', 'high'), ('play', 'no') Coverage: 4
Item set: ('humidity', 'high'), ('windy', 'FALSE') Coverage: 4
Item set: ('humidity', 'normal'), ('windy', 'FALSE') Coverage: 4
Item set: ('humidity', 'high'), ('temperature', 'mild') Coverage: 4
Item set: ('humidity', 'normal'), ('temperature', 'cool') Coverage: 4
Item set: ('humidity', 'normal'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 4

Reporting back all association rules that meet requirements:

Rule: if ('humidity', 'normal'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 1.0
Rule: if ('temperature', 'cool') then ('humidity', 'normal') Confidence: 1.0
Rule: if ('outlook', 'overcast') then ('play', 'yes') Confidence: 1.0
Rule: if ('humidity', 'normal') then ('play', 'yes') Confidence: 0.857142857143
Rule: if ('play', 'no') then ('humidity', 'high') Confidence: 0.8
Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75
Rule: if ('humidity', 'normal'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('humidity', 'normal') Confidence: 0.666666666667
Rule: if ('temperature', 'mild') then ('humidity', 'high') Confidence: 0.666666666667
Rule: if ('play', 'yes') then ('humidity', 'normal') Confidence: 0.666666666667
Rule: if ('temperature', 'mild') then ('play', 'yes') Confidence: 0.666666666667
Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if _ then ('play', 'yes') Confidence: 0.642857142857
Rule: if ('humidity', 'normal') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('humidity', 'normal') then ('temperature', 'cool') Confidence: 0.571428571429
Rule: if ('humidity', 'high') then ('temperature', 'mild') Confidence: 0.571428571429
Rule: if ('humidity', 'normal') then ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('humidity', 'high') then ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('humidity', 'high') then ('play', 'no') Confidence: 0.571428571429
Rule: if _ then ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('windy', 'FALSE') then ('humidity', 'normal'), ('play', 'yes') Confidence: 0.5
Rule: if ('windy', 'FALSE') then ('humidity', 'normal') Confidence: 0.5
Rule: if ('windy', 'FALSE') then ('humidity', 'high') Confidence: 0.5
Rule: if _ then ('humidity', 'normal') Confidence: 0.5
Rule: if _ then ('humidity', 'high') Confidence: 0.5

TEST #15

Run Apriori on large dataset iris.arff with numeric values showing 6 rules

r08lnk7x9:CS 5001 Lauren\$ python apriori.py

Name of the input file: iris.arff

Minimum coverage (1-n): 10

Maximum size of sets to consider (1-n): 4

Minimum accuracy (0.0-1.0): 0.5

Number of rules to report back (1-n, 'all'): 6

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (4):

Item set: ('sepalwidth', '5.0') Coverage: 10
Item set: ('petalwidth', '0.2') Coverage: 28
Item set: ('petalwidth', '1.5') Coverage: 12
Item set: ('petalwidth', '1.3') Coverage: 13
Item set: ('petalwidth', '1.8') Coverage: 12
Item set: ('sepalwidth', '3.0') Coverage: 26
Item set: ('sepalwidth', '3.2') Coverage: 13
Item set: ('sepalwidth', '3.1') Coverage: 12
Item set: ('sepalwidth', '3.4') Coverage: 12
Item set: ('sepalwidth', '2.9') Coverage: 10
Item set: ('sepalwidth', '2.8') Coverage: 14
Item set: ('class', 'Iris-setosa') Coverage: 50
Item set: ('class', 'Iris-versicolor') Coverage: 50
Item set: ('class', 'Iris-virginica') Coverage: 50
Item set: ('petallength', '1.4') Coverage: 12
Item set: ('petallength', '1.5') Coverage: 14
Item set: ('class', 'Iris-setosa'), ('petalwidth', '0.2') Coverage: 28
Item set: ('class', 'Iris-versicolor'), ('petalwidth', '1.5') Coverage: 10
Item set: ('class', 'Iris-versicolor'), ('petalwidth', '1.3') Coverage: 13
Item set: ('class', 'Iris-virginica'), ('petalwidth', '1.8') Coverage: 11
Item set: ('class', 'Iris-virginica'), ('sepalwidth', '3.0') Coverage: 12
Item set: ('class', 'Iris-setosa'), ('petallength', '1.4') Coverage: 12
Item set: ('class', 'Iris-setosa'), ('petallength', '1.5') Coverage: 14

Reporting back the most accurate 6 association rules that meet requirements:

Rule: if ('petallength', '1.5') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petallength', '1.4') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petalwidth', '1.3') then ('class', 'Iris-versicolor') Confidence: 1.0
Rule: if ('petalwidth', '0.2') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petalwidth', '1.8') then ('class', 'Iris-virginica') Confidence: 0.916666666667
Rule: if ('petalwidth', '1.5') then ('class', 'Iris-versicolor') Confidence: 0.833333333333

TEST #16

Run Apriori on small dataset weather.arff with numeric values showing all rules

r08lnk7x9:CS 5001 Lauren\$ python apriori.py

Name of the input file: weather.arff

Minimum coverage (1-n): 3

Maximum size of sets to consider (1-n): 4

Minimum accuracy (0.0-1.0): 0.5

Number of rules to report back (1-n, 'all'): all

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (4):

Item set: ('play', 'yes') Coverage: 9
Item set: ('play', 'no') Coverage: 5
Item set: ('windy', 'TRUE') Coverage: 6
Item set: ('windy', 'FALSE') Coverage: 8
Item set: ('outlook', 'sunny') Coverage: 5
Item set: ('outlook', 'overcast') Coverage: 4
Item set: ('outlook', 'rainy') Coverage: 5
Item set: ('humidity', '70') Coverage: 3
Item set: ('play', 'yes'), ('windy', 'TRUE') Coverage: 3
Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6
Item set: ('outlook', 'overcast'), ('play', 'yes') Coverage: 4
Item set: ('outlook', 'rainy'), ('play', 'yes') Coverage: 3
Item set: ('play', 'no'), ('windy', 'TRUE') Coverage: 3
Item set: ('outlook', 'sunny'), ('play', 'no') Coverage: 3
Item set: ('outlook', 'sunny'), ('windy', 'FALSE') Coverage: 3
Item set: ('outlook', 'rainy'), ('windy', 'FALSE') Coverage: 3
Item set: ('outlook', 'rainy'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 3

Reporting back all association rules that meet requirements:

Rule: if ('outlook', 'rainy'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 1.0
Rule: if ('outlook', 'rainy'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 1.0
Rule: if ('outlook', 'overcast') then ('play', 'yes') Confidence: 1.0
Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75
Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.6666666666666667
Rule: if _ then ('play', 'yes') Confidence: 0.642857142857
Rule: if ('outlook', 'rainy') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'rainy') then ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'sunny') then ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'sunny') then ('play', 'no') Confidence: 0.6
Rule: if ('play', 'no') then ('outlook', 'sunny') Confidence: 0.6
Rule: if ('play', 'no') then ('windy', 'TRUE') Confidence: 0.6
Rule: if ('outlook', 'rainy') then ('play', 'yes') Confidence: 0.6
Rule: if _ then ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('outlook', 'rainy') Confidence: 0.5
Rule: if ('windy', 'TRUE') then ('play', 'no') Confidence: 0.5
Rule: if ('windy', 'TRUE') then ('play', 'yes') Confidence: 0.5

TEST #17

Run Apriori on weather.arff to show user input variations:

r08lnk7x9:CS 5001 Lauren\$ python apriori.py

Name of the input file: weather.arff

Minimum coverage (1-n): 3

Maximum size of sets to consider (1-n): 3

Minimum accuracy (0.0-1.0): 0.4

Number of rules to report back (1-n, 'all'): all

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (3):

Item set: ('play', 'yes') Coverage: 9
Item set: ('play', 'no') Coverage: 5
Item set: ('windy', 'TRUE') Coverage: 6
Item set: ('windy', 'FALSE') Coverage: 8
Item set: ('outlook', 'sunny') Coverage: 5
Item set: ('outlook', 'overcast') Coverage: 4
Item set: ('outlook', 'rainy') Coverage: 5
Item set: ('humidity', '70') Coverage: 3
Item set: ('play', 'yes'), ('windy', 'TRUE') Coverage: 3
Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6
Item set: ('outlook', 'overcast'), ('play', 'yes') Coverage: 4
Item set: ('outlook', 'rainy'), ('play', 'yes') Coverage: 3
Item set: ('play', 'no'), ('windy', 'TRUE') Coverage: 3
Item set: ('outlook', 'sunny'), ('play', 'no') Coverage: 3
Item set: ('outlook', 'sunny'), ('windy', 'FALSE') Coverage: 3
Item set: ('outlook', 'rainy'), ('windy', 'FALSE') Coverage: 3
Item set: ('outlook', 'rainy'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 3

Reporting back all association rules that meet requirements:

Rule: if ('outlook', 'rainy'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 1.0
Rule: if ('outlook', 'rainy'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 1.0
Rule: if ('outlook', 'overcast') then ('play', 'yes') Confidence: 1.0
Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75
Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.6666666666667
Rule: if _ then ('play', 'yes') Confidence: 0.642857142857
Rule: if ('outlook', 'rainy') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'rainy') then ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'sunny') then ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'sunny') then ('play', 'no') Confidence: 0.6
Rule: if ('play', 'no') then ('outlook', 'sunny') Confidence: 0.6
Rule: if ('play', 'no') then ('windy', 'TRUE') Confidence: 0.6
Rule: if ('outlook', 'rainy') then ('play', 'yes') Confidence: 0.6
Rule: if _ then ('windy', 'FALSE') Confidence: 0.571428571429

Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('outlook', 'rainy') Confidence: 0.5
Rule: if ('windy', 'TRUE') then ('play', 'no') Confidence: 0.5
Rule: if ('windy', 'TRUE') then ('play', 'yes') Confidence: 0.5
Rule: if ('play', 'yes') then ('outlook', 'overcast') Confidence: 0.444444444444
Rule: if _ then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.428571428571
Rule: if _ then ('windy', 'TRUE') Confidence: 0.428571428571

TEST #18

Run Apriori on weather-nominal.arff to show user input variations:

r08lnk7x9:CS 5001 Lauren\$ python apriori.py

Name of the input file: weather-nominal.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 1
Minimum accuracy (0.0-1.0): 0.4
Number of rules to report back (1-n, 'all'): 5

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (1):

Item set: ('play', 'yes') Coverage: 9
Item set: ('play', 'no') Coverage: 5
Item set: ('windy', 'TRUE') Coverage: 6
Item set: ('windy', 'FALSE') Coverage: 8
Item set: ('outlook', 'sunny') Coverage: 5
Item set: ('outlook', 'overcast') Coverage: 4
Item set: ('outlook', 'rainy') Coverage: 5
Item set: ('temperature', 'hot') Coverage: 4
Item set: ('temperature', 'mild') Coverage: 6
Item set: ('temperature', 'cool') Coverage: 4
Item set: ('humidity', 'high') Coverage: 7
Item set: ('humidity', 'normal') Coverage: 7

Reporting back the most accurate 5 association rules that meet requirements:

Rule: if _ then ('play', 'yes') Confidence: 0.642857142857
Rule: if _ then ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if _ then ('humidity', 'normal') Confidence: 0.5
Rule: if _ then ('humidity', 'high') Confidence: 0.5
Rule: if _ then ('temperature', 'mild') Confidence: 0.428571428571

FP Growth Algorithm

TEST #1

Give file not of type .arff

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: readme.txt  
The file must be of type .arff
```

TEST #2

Give min coverage < 1

```
Bastion-2:cs5001 $ python FPgrowth.py
```

```
Name of the input file: weather.arff  
Minimum coverage (1-n): 0  
Min coverage should be 1 or more, try again:  
Minimum coverage (1-n):
```

TEST #3

Give min coverage = 1

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff  
Minimum coverage (1-n): 1  
Maximum size of sets to consider (1-n):
```

TEST #4

Give min coverage a string value

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff  
Minimum coverage (1-n): string  
Min coverage should be 1 or more, try again:  
Minimum coverage (1-n):
```

TEST #5

Give max size < 0

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): -1
Max size should be 1 or more, try again:
Maximum size of sets to consider (1-n):
```

TEST #6

Give max size = 0

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 0
Max size should be 1 or more, try again:
Maximum size of sets to consider (1-n):
```

TEST #7

Give max size a string value

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): string
Max size should be 1 or more, try again:
Maximum size of sets to consider (1-n):
```

TEST #8

Give min accuracy < 0

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): -1
Min accuracy should be between 0-1, try again:
Minimum accuracy (0.0-1.0):
```

TEST #9

Give min accuracy > 1

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 2
Min accuracy should be between 0-1, try again:
Minimum accuracy (0.0-1.0):
```

TEST #10

Give min accuracy a string value

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): string
Min accuracy should be between 0-1, try again:
Minimum accuracy (0.0-1.0):
```

TEST #11

Give report number < 0

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): -1
Report number should be an integer above 0 or 'all', try
again:
Number of rules to report back (1-n, 'all'):
```

TEST #12

Give report number = 0

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): 0
Report number should be an integer above 0 or 'all', try
again:
Number of rules to report back (1-n, 'all'):
```

TEST #13

Give report number a string value other than 'all'

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): string
Report number should be an integer above 0 or 'all', try
again:
Number of rules to report back (1-n, 'all'):
```

TEST #14

Run FP Growth on small dataset weather-nominal.arff (nominal values) showing all rules

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

```
Name of the input file: weather-nominal.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): all
```

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (4):

```
Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6
Item set: ('humidity', 'normal'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 4
```

Reporting back all association rules that meet requirements:

```
Rule: if ('humidity', 'normal'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 1.0
Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75
```

Rule: if ('humidity', 'normal'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('humidity', 'normal') Confidence: 0.666666666667
Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('humidity', 'normal') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('windy', 'FALSE') then ('humidity', 'normal'), ('play', 'yes') Confidence: 0.5

TEST #15

Run FP Growth on large dataset iris.arff (has numeric values) showing 15 rules

r08lnk7x9:CS 5001 Lauren\$ python FPgrowth.py

Name of the input file: iris.arff
Minimum coverage (1-n): 4
Maximum size of sets to consider (1-n): 5
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): 15

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (5):

Item set: ('class', 'Iris-setosa'), ('petalwidth', '0.2') Coverage: 28
Item set: ('class', 'Iris-setosa'), ('petallength', '1.5') Coverage: 14
Item set: ('class', 'Iris-versicolor'), ('petalwidth', '1.3') Coverage: 13
Item set: ('class', 'Iris-versicolor'), ('sepalwidth', '3.0') Coverage: 8
Item set: ('class', 'Iris-versicolor'), ('sepalwidth', '2.8') Coverage: 6
Item set: ('class', 'Iris-versicolor'), ('sepalwidth', '2.7') Coverage: 5
Item set: ('class', 'Iris-virginica'), ('sepalwidth', '2.8') Coverage: 8
Item set: ('class', 'Iris-virginica'), ('sepalwidth', '3.0') Coverage: 12
Item set: ('class', 'Iris-virginica'), ('petalwidth', '1.8') Coverage: 11
Item set: ('class', 'Iris-virginica'), ('sepalwidth', '3.2') Coverage: 5
Item set: ('class', 'Iris-virginica'), ('sepalwidth', '3.1') Coverage: 4
Item set: ('class', 'Iris-setosa'), ('petallength', '1.5'), ('petalwidth', '0.2') Coverage: 6
Item set: ('class', 'Iris-setosa'), ('petalwidth', '0.2'), ('sepalwidth', '3.4') Coverage: 6
Item set: ('class', 'Iris-setosa'), ('petallength', '1.4'), ('petalwidth', '0.2') Coverage: 8
Item set: ('class', 'Iris-setosa'), ('petalwidth', '0.2'), ('sepalwidth', '3.2') Coverage: 5
Item set: ('class', 'Iris-versicolor'), ('petalwidth', '1.3'), ('sepalwidth', '2.9') Coverage: 5
Item set: ('class', 'Iris-virginica'), ('petalwidth', '1.8'), ('sepalwidth', '3.0') Coverage: 4

Reporting back the most accurate 15 association rules that meet requirements:

Rule: if ('petalwidth', '1.8'), ('sepalwidth', '3.0') then ('class', 'Iris-virginica') Confidence: 1.0
Rule: if ('petalwidth', '1.3'), ('sepalwidth', '2.9') then ('class', 'Iris-versicolor') Confidence: 1.0
Rule: if ('class', 'Iris-setosa'), ('sepalwidth', '3.2') then ('petalwidth', '0.2') Confidence: 1.0
Rule: if ('petalwidth', '0.2'), ('sepalwidth', '3.2') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petallength', '1.4'), ('petalwidth', '0.2') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petalwidth', '0.2'), ('sepalwidth', '3.4') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petallength', '1.5'), ('petalwidth', '0.2') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petalwidth', '1.3') then ('class', 'Iris-versicolor') Confidence: 1.0
Rule: if ('petallength', '1.5') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petalwidth', '0.2') then ('class', 'Iris-setosa') Confidence: 1.0
Rule: if ('petalwidth', '1.8') then ('class', 'Iris-virginica') Confidence: 0.916666666667
Rule: if ('petalwidth', '1.8') then ('class', 'Iris-virginica') Confidence: 0.916666666667

Rule: if ('class', 'Iris-versicolor'), ('sepalwidth', '2.9') then ('petalwidth', '1.3') Confidence: 0.714285714286
Rule: if ('petallength', '1.4') then ('class', 'Iris-setosa'), ('petalwidth', '0.2') Confidence: 0.666666666667
Rule: if ('class', 'Iris-setosa'), ('petallength', '1.4') then ('petalwidth', '0.2') Confidence: 0.666666666667

TEST #16

Run FP Growth on small dataset weather.arff with numeric values showing all rules

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

Name of the input file: weather.arff
Minimum coverage (1-n): 2
Maximum size of sets to consider (1-n): 4
Minimum accuracy (0.0-1.0): 0.5
Number of rules to report back (1-n, 'all'): all

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (4):

Item set: ('outlook', 'sunny'), ('windy', 'FALSE') Coverage: 3
Item set: ('outlook', 'rainy'), ('windy', 'TRUE') Coverage: 2
Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6
Item set: ('play', 'yes'), ('windy', 'TRUE') Coverage: 3
Item set: ('outlook', 'rainy'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 3

Reporting back all association rules that meet requirements:

Rule: if ('outlook', 'rainy'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 1.0
Rule: if ('outlook', 'rainy'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 1.0
Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75
Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('outlook', 'rainy') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.6
Rule: if ('outlook', 'sunny') then ('windy', 'FALSE') Confidence: 0.6
Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('outlook', 'rainy') Confidence: 0.5
Rule: if ('windy', 'TRUE') then ('play', 'yes') Confidence: 0.5

TEST #17

Run FP Growth on weather.arff to show user input variations:

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

Name of the input file: weather.arff

Minimum coverage (1-n): 2

Maximum size of sets to consider (1-n): 2

Minimum accuracy (0.0-1.0): 0.3

Number of rules to report back (1-n, 'all'): 7

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (2):

Item set: ('outlook', 'sunny'), ('windy', 'FALSE') Coverage: 3

Item set: ('outlook', 'rainy'), ('windy', 'TRUE') Coverage: 2

Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6

Item set: ('play', 'yes'), ('windy', 'TRUE') Coverage: 3

Reporting back the most accurate 7 association rules that meet requirements:

Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75

Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.66666666666667

Rule: if ('outlook', 'sunny') then ('windy', 'FALSE') Confidence: 0.6

Rule: if ('windy', 'TRUE') then ('play', 'yes') Confidence: 0.5

Rule: if _ then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.428571428571

Rule: if ('outlook', 'rainy') then ('windy', 'TRUE') Confidence: 0.4

Rule: if ('windy', 'FALSE') then ('outlook', 'sunny') Confidence: 0.375

TEST #18

Run FP Growth on weather-nominal.arff to show input variations:

```
r08lnk7x9:CS 5001 Lauren$ python FPgrowth.py
```

Name of the input file: weather-nominal.arff

Minimum coverage (1-n): 2

Maximum size of sets to consider (1-n): 3

Minimum accuracy (0.0-1.0): 0.25

Number of rules to report back (1-n, 'all'): all

All sets that meet Minimum Coverage and are no more than the Max Size of Item Sets (3):

Item set: ('play', 'yes'), ('windy', 'FALSE') Coverage: 6

Item set: ('humidity', 'normal'), ('play', 'yes') Coverage: 6

Item set: ('humidity', 'high'), ('windy', 'FALSE') Coverage: 4

Item set: ('humidity', 'high'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 2

Item set: ('humidity', 'normal'), ('play', 'yes'), ('windy', 'FALSE') Coverage: 4

Reporting back all association rules that meet requirements:

Rule: if ('humidity', 'normal'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 1.0
Rule: if ('humidity', 'normal') then ('play', 'yes') Confidence: 0.857142857143
Rule: if ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.75
Rule: if ('humidity', 'normal'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('humidity', 'normal') Confidence: 0.666666666667
Rule: if ('humidity', 'high'), ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('play', 'yes') then ('humidity', 'normal') Confidence: 0.666666666667
Rule: if ('play', 'yes') then ('windy', 'FALSE') Confidence: 0.666666666667
Rule: if ('humidity', 'normal') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('humidity', 'high') then ('windy', 'FALSE') Confidence: 0.571428571429
Rule: if ('windy', 'FALSE') then ('humidity', 'normal'), ('play', 'yes') Confidence: 0.5
Rule: if ('humidity', 'high'), ('windy', 'FALSE') then ('play', 'yes') Confidence: 0.5
Rule: if ('windy', 'FALSE') then ('humidity', 'high') Confidence: 0.5
Rule: if ('play', 'yes') then ('humidity', 'normal'), ('windy', 'FALSE') Confidence: 0.444444444444
Rule: if _ then ('humidity', 'normal'), ('play', 'yes') Confidence: 0.428571428571
Rule: if _ then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.428571428571
Rule: if ('play', 'yes'), ('windy', 'FALSE') then ('humidity', 'high') Confidence: 0.333333333333
Rule: if ('humidity', 'high') then ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.285714285714
Rule: if _ then ('humidity', 'normal'), ('play', 'yes'), ('windy', 'FALSE') Confidence: 0.285714285714
Rule: if _ then ('humidity', 'high'), ('windy', 'FALSE') Confidence: 0.285714285714
Rule: if ('windy', 'FALSE') then ('humidity', 'high'), ('play', 'yes') Confidence: 0.25