Project 1:
Dimensionality Reduction and Association Analysis

Aswin Shakil Balasubramanian	Sanidhya Chopde	Shashank Raghunathan
UB Name: aswinsha	UB Name: schopde	UB Name: raghuna2

Part 2: Association Analysis

Apriori Algorithm: Apriori is an algorithm for frequent item set mining and association rule learning. It involves identifying the frequent individual items and extending them to larger item sets.

It follows two ideas:

1. All subsets of a frequent itemset must be frequent.
2. If an itemset is infrequent, all of its supersets will be infrequent.

Algorithm flow:

Our code follows the following process –

1. Frequent itemset generation
2. Rule generation
3. Template queries

Frequent Itemset Generation:

1. Take the support threshold as input from the user.
2. We start by reading the input file – "association-rule-test-data.txt" and doing some preprocessing. In the pre preprocessing, we add "G" followed by the column number before each gene which is either an "Up/Down". Eg: G59_Up.
3. Along with adding the Gs, we also store the number of unique itemsets including their count and store them in a dictionary. i.e itemset of length 1.
4. We now generate the candidate items for length 2. To do so, we will take the input of the frequent itemset of the previous length. i.e length-1.
5. We now have the candidate itemset for length 2 and need to generate the frequent itemset for length 2.
6. We increase the length by 1 in every subsequent step until there are no new frequent items of length n.
7. In each step, we also prune the itemset based on the support value.
8. We update the dictionary for every length. i.e Dictionary[1] has all frequent items with length 1, dictionary[2] has all the frequent items with length 2 and and so on.

Rule generation:

1. Take the confidence threshold from the user as input.
2. If we have a frequent itemset of length 'n' which is the max length, we create all the rules with length 'n-1' to 1 as the head.
3. Eg: If we have {A,B,C} as a frequent itemset, we can generate rules like:
    a. {A,B} => {C}
    b. {A,C} => {B}
    c. {C,B} => {A}
    d. {A} => {C,B}
    e. {B} => {A,C}
    f. {C} => {A,B}
4. After generating the $2^k - 2$ candidate association rules, we select the ones which have a confidence value greater than the threshold we have taken as input.

Template Queries: We have been given 3 formats for the template queries.

1. **Template 1**: In this format, we take in 3 inputs from the user,
    i. The first parameter can take (RULE, HEAD, BODY) as input.
    ii. The second parameter can take (ANY, NONE, Number) as input.
    iii. The third parameter can take an itemset separated by commas as input.
    iv. The final rules set are generated using our template1 helper method.
2. **Template 2:** In this format, we take in 2 inputs from the user,
    i. The first parameter can take (RULE, HEAD, BODY) as input.
    ii. The second parameter can take a number as an input.
    iii. The final rules set are generated using our template2 helper method.
3. **Template 3:** In this format, we take one input from the user which can be of 6(six) types. 1or1; 1or2; 2or2; 1and1;1and2; 2and2;
    i. Based on the input type, we perform template 1 or template 2 query formats with or(union) / and(intersection) conjunction operations.

Output:

Part 1:

```
Enter Support Threshold: 30
The number of frequent items of length 1 are:  196
The number of frequent items of length 2 are:  5340
The number of frequent items of length 3 are:  5287
The number of frequent items of length 4 are:  1518
The number of frequent items of length 5 are:  438
The number of frequent items of length 6 are:  88
The number of frequent items of length 7 are:  11
The number of frequent items of length 8 are:  1
The number of frequent items of length 9 are:  0
The number of all length frequent items are:  12879
```

```
Enter Support Threshold: 40
The number of frequent items of length 1 are:  167
The number of frequent items of length 2 are:  753
The number of frequent items of length 3 are:  149
The number of frequent items of length 4 are:  7
The number of frequent items of length 5 are:  1
The number of frequent items of length 6 are:  0
The number of all length frequent items are:  1077

Enter Support Threshold: 50
The number of frequent items of length 1 are:  109
The number of frequent items of length 2 are:  63
The number of frequent items of length 3 are:  2
The number of frequent items of length 4 are:  0
The number of all length frequent items are:  174

Enter Support Threshold: 60
The number of frequent items of length 1 are:  34
The number of frequent items of length 2 are:  2
The number of frequent items of length 3 are:  0
The number of all length frequent items are:  36

Enter Support Threshold: 70
The number of frequent items of length 1 are:  7
The number of frequent items of length 2 are:  0
The number of all length frequent items are:  7
```

Part 2:
Template 1:

(result11, cnt) = asso_rule.template1("RULE", "ANY", ['G59_UP'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: rule
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 1 is 26
```

(result12, cnt) = asso_rule.template1("RULE", "NONE", ['G59_UP'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: rule
Enter the second parameter - ANY|NONE|NUMBER: none
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 1 is 91
```

(result13, cnt) = asso_rule.template1("RULE", 1, ['G59_UP', 'G10_Down'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: rule
Enter the second parameter - ANY|NONE|NUMBER: 1
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up,G10_Down
The number of rules for template 1 is 39
```

(result14, cnt) = asso_rule.template1("HEAD", "ANY", ['G59_UP'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 1 is 9
```

(result15, cnt) = asso_rule.template1("HEAD", "NONE", ['G59_UP'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - ANY|NONE|NUMBER: none
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 1 is 108
```

(result16, cnt) = asso_rule.template1("HEAD", 1, ['G59_UP', 'G10_Down'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: Head
Enter the second parameter - ANY|NONE|NUMBER: 1
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up,G10_Down
The number of rules for template 1 is 17
```

(result17, cnt) = asso_rule.template1("BODY", "ANY", ['G59_UP'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 1 is 17
```

(result18, cnt) = asso_rule.template1("BODY", "NONE", ['G59_UP'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - ANY|NONE|NUMBER: none
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 1 is 100
```

(result19, cnt) = asso_rule.template1("BODY", 1, ['G59_UP', 'G10_Down'])

```
Enter the Template number: 1
Enter the first parameter - RULE|HEAD|BODY: Body
Enter the second parameter - ANY|NONE|NUMBER: 1
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up,G10_Down
The number of rules for template 1 is 24
```

Template 2:

(result21, cnt) = asso_rule.template2("RULE", 3)

```
Enter the Template number: 2
Enter the first parameter - RULE|HEAD|BODY: Rule
Enter the second parameter - Any valid number: 3
The number of rules for template 2 is 9
```

(result22, cnt) = asso_rule.template2("HEAD", 2)

```
Enter the Template number: 2
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - Any valid number: 2
The number of rules for template 2 is 6
```

(result23, cnt) = asso_rule.template2("BODY", 1)

```
Enter the Template number: 2
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - Any valid number: 1
The number of rules for template 2 is 117
```

Template 3:

(result31, cnt) = asso_rule.template3("1or1", "HEAD", "ANY", ['G10_Down'], "BODY", 1, ['G59_UP'])

```
Enter the Template number: 3
Enter the paramater for template 3: 1or1
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G10_Down
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - ANY|NONE|NUMBER: 1
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 3 is 24
```

(result32, cnt) = asso_rule.template3("1and1", "HEAD", "ANY", ['G10_Down'], "BODY", 1, ['G59_UP'])

```
Enter the Template number: 3
Enter the paramater for template 3: 1and1
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G10_Down
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - ANY|NONE|NUMBER: 1
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G59_Up
The number of rules for template 3 is 1
```

(result33, cnt) = asso_rule.template3("1or2", "HEAD", "ANY", ['G10_Down'], "BODY", 2)

```
Enter the Template number: 3
Enter the paramater for template 3: 1or2
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G10_Down
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - Any valid number: 2
The number of rules for template 3 is 11
```

(result34, cnt) = asso_rule.template3("1and2", "HEAD", "ANY", ['G10_Down'], "BODY", 2)

```
Enter the Template number: 3
Enter the paramater for template 3: 1and2
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - ANY|NONE|NUMBER: any
Enter the itemset - ITEM1,ITEM2,..,ITEMN: G10_Down
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - Any valid number: 2
The number of rules for template 3 is 0
```

(result35, cnt) = asso_rule.template3("2or2", "HEAD", 1, "BODY", 2)

```
Enter the Template number: 3
Enter the paramater for template 3: 2or2
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - Any valid number: 1
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - Any valid number: 2
The number of rules for template 3 is 117
```

(result36, cnt) = asso_rule.template3("2and2", "HEAD", 1, "BODY", 2)

```
Enter the Template number: 3
Enter the paramater for template 3: 2and2
Enter the first parameter - RULE|HEAD|BODY: head
Enter the second parameter - Any valid number: 1
Enter the first parameter - RULE|HEAD|BODY: body
Enter the second parameter - Any valid number: 2
The number of rules for template 3 is 3
```