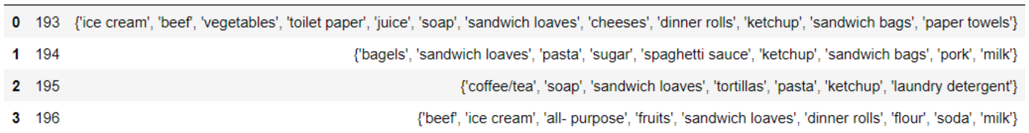
Report

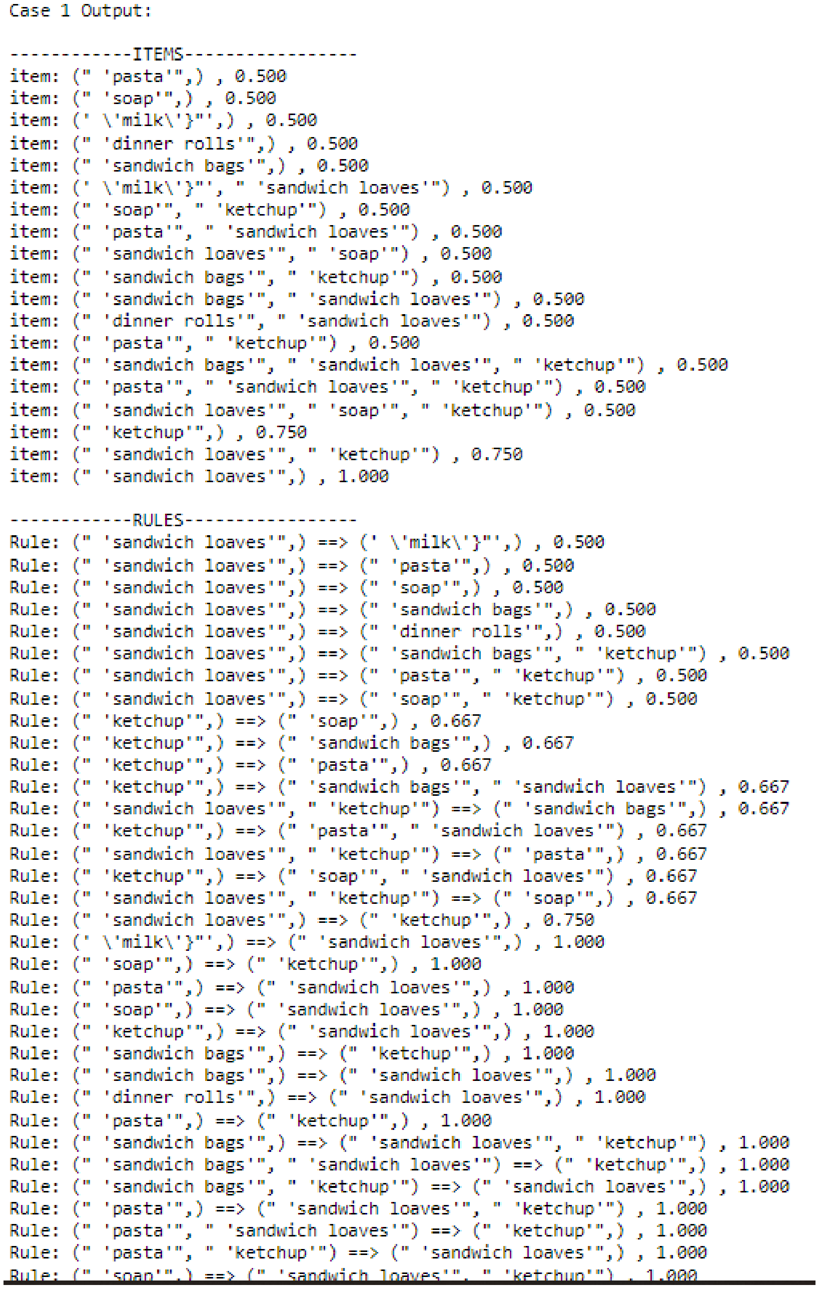
For the given dataset we extract 4 rows and create a sample dataset to evaluate the results using apriori algorithm. In This case we take the 4 rows from 192 to 196 and Run the algorithm



0	193	{'ice cream', 'beef', 'vegetables', 'toilet paper', 'juice', 'soap', 'sandwich loaves', 'cheeses', 'dinner rolls', 'ketchup', 'sandwich bags', 'paper towels'}
1	194	{'bagels', 'sandwich loaves', 'pasta', 'sugar', 'spaghetti sauce', 'ketchup', 'sandwich bags', 'pork', 'milk'}
2	195	{'coffee/tea', 'soap', 'sandwich loaves', 'tortillas', 'pasta', 'ketchup', 'laundry detergent'}
3	196	{'beef', 'ice cream', 'all- purpose', 'fruits', 'sandwich loaves', 'dinner rolls', 'flour', 'soda', 'milk'}

Case1:

For minimum support = 0.26 and minimum confidence = 0.5 we get



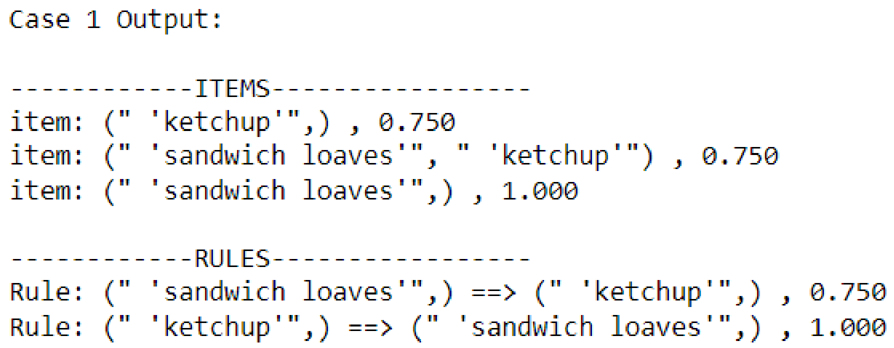
```
Case 1 Output:

------------ITEMS-----------------
item: (" 'pasta'",) , 0.500
item: (" 'soap'",) , 0.500
item: (' \'milk\'}"',) , 0.500
item: (" 'dinner rolls'",) , 0.500
item: (" 'sandwich bags'",) , 0.500
item: (' \'milk\'}"', " 'sandwich loaves'") , 0.500
item: (" 'soap'", " 'ketchup'") , 0.500
item: (" 'pasta'", " 'sandwich loaves'") , 0.500
item: (" 'sandwich loaves'", " 'soap'") , 0.500
item: (" 'sandwich bags'", " 'ketchup'") , 0.500
item: (" 'sandwich bags'", " 'sandwich loaves'") , 0.500
item: (" 'dinner rolls'", " 'sandwich loaves'") , 0.500
item: (" 'pasta'", " 'ketchup'") , 0.500
item: (" 'sandwich bags'", " 'sandwich loaves'", " 'ketchup'") , 0.500
item: (" 'pasta'", " 'sandwich loaves'", " 'ketchup'") , 0.500
item: (" 'sandwich loaves'", " 'soap'", " 'ketchup'") , 0.500
item: (" 'ketchup'",) , 0.750
item: (" 'sandwich loaves'", " 'ketchup'") , 0.750
item: (" 'sandwich loaves'",) , 1.000

------------RULES-----------------
Rule: (" 'sandwich loaves'",) ==> (' \'milk\'}"',) , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'pasta'",) , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'soap'",) , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'sandwich bags'",) , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'dinner rolls'",) , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'sandwich bags'", " 'ketchup'") , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'pasta'", " 'ketchup'") , 0.500
Rule: (" 'sandwich loaves'",) ==> (" 'soap'", " 'ketchup'") , 0.500
Rule: (" 'ketchup'",) ==> (" 'soap'",) , 0.667
Rule: (" 'ketchup'",) ==> (" 'sandwich bags'",) , 0.667
Rule: (" 'ketchup'",) ==> (" 'pasta'",) , 0.667
Rule: (" 'ketchup'",) ==> (" 'sandwich bags'", " 'sandwich loaves'") , 0.667
Rule: (" 'sandwich loaves'", " 'ketchup'") ==> (" 'sandwich bags'",) , 0.667
Rule: (" 'ketchup'",) ==> (" 'pasta'", " 'sandwich loaves'") , 0.667
Rule: (" 'sandwich loaves'", " 'ketchup'") ==> (" 'pasta'",) , 0.667
Rule: (" 'ketchup'",) ==> (" 'soap'", " 'sandwich loaves'") , 0.667
Rule: (" 'sandwich loaves'", " 'ketchup'") ==> (" 'soap'",) , 0.667
Rule: (" 'sandwich loaves'",) ==> (" 'ketchup'",) , 0.750
Rule: (' \'milk\'}"',) ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'soap'",) ==> (" 'ketchup'",) , 1.000
Rule: (" 'pasta'",) ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'soap'",) ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'ketchup'",) ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'sandwich bags'",) ==> (" 'ketchup'",) , 1.000
Rule: (" 'sandwich bags'",) ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'dinner rolls'",) ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'pasta'",) ==> (" 'ketchup'",) , 1.000
Rule: (" 'sandwich bags'",) ==> (" 'sandwich loaves'", " 'ketchup'") , 1.000
Rule: (" 'sandwich bags'", " 'sandwich loaves'") ==> (" 'ketchup'",) , 1.000
Rule: (" 'sandwich bags'", " 'ketchup'") ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'pasta'",) ==> (" 'sandwich loaves'", " 'ketchup'") , 1.000
Rule: (" 'pasta'", " 'sandwich loaves'") ==> (" 'ketchup'",) , 1.000
Rule: (" 'pasta'", " 'ketchup'") ==> (" 'sandwich loaves'",) , 1.000
Rule: (" 'soap'",) ==> (" 'sandwich loaves'", " 'ketchup'") , 1.000
```

- Since we have given minimum support as 0.26 the most frequent items generated have support more than 26% hence we get the item sets which have support 0.26 and above
- As earlier mentioned support will be calculated as
  - **Support(i(1st item)) =** Item’s frequency/total number of transactions
  - **Example:** item: (" 'sandwich loaves'", " 'soap'", " 'ketchup'") , 0.500 with three item set has minimum support more than 0.26
- Since the minimum confidence is 70% the non-empty subsets are calculated which have confidence intervals more than 70%
  - For example, Rule: (" 'sandwich loaves'",) ==> (" 'ketchup'",) has a confidence interval of 75% (>50%) hence the rule is generated

Reasoning:
According to Apriori algorithm, if we keep the support count to minimum, the more rules it produces. As we can see when we put the minimum support as 0.26 which is 26% and minimum confidence to 0.5 which is 50%, we get large number of rules.

Case2:

In This case we use minimum support = 0.51 and minimum confidence = 0.55 we get

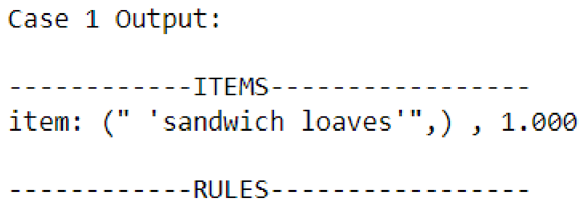


```
Case 1 Output:

------------ITEMS-----------------
item: (" 'ketchup'",) , 0.750
item: (" 'sandwich loaves'", " 'ketchup'") , 0.750
item: (" 'sandwich loaves'",) , 1.000

------------RULES-----------------
Rule: (" 'sandwich loaves'",) ==> (" 'ketchup'",) , 0.750
Rule: (" 'ketchup'",) ==> (" 'sandwich loaves'",) , 1.000
```

- Considering we have given minimum support as 0.51 the most frequent items generated have support more than 51% hence we get the item sets which have support 0.5 and above and in this case we have 3 items
  - **Support(i(1st item)) =** Item’s frequency/total number of transactions
  - **Example:** item: (" 'sandwich loaves'", " 'ketchup'") , 0.750 with three item set has minimum support more than 0.51
- Since the minimum confidence is 55% the non-empty subsets are calculated which have confidence intervals more than 55%
  - **Confidence** = support of itemset/support(i(1st item))
  - **Example:** Rule: (" 'sandwich loaves'",) ==> (" 'ketchup'",) , 0.750 with a 1-itemset has support more than 55%

Reasoning: In case 2, we tried to increase the minimum support to 0.51 and minimum confidence to 0.55, thereby decreasing the rules when compared to case 1.
Moreover, it would be very challenging to analyze the data and background calculations for larger datasets, so we kept the reasoning to understanding the working functionality of algorithm.

Case3:
In This case we use minimum support = 0.76 and minimum confidence = 0.6 we get



```
Case 1 Output:

------------ITEMS-----------------
item: (" 'sandwich loaves'",) , 1.000

------------RULES-----------------
```

- Considering we have given minimum support as 0.76 the most frequent items generated have support more than 60% hence we get the item sets which have support 0.76 and above and in this case we have 1 items
    - **Support(i(1st item)) =** Item’s frequency/total number of transactions
    - **Example:** item: (" 'sandwich loaves'",) , 1.000 with three item set has minimum support more than 0.76
- Since the minimum confidence is 60% there are no rules calculated for the given non empty subsets for the item sets hence none are returned

Reasoning:
After observing the support counts of each item in the given dataset we concluded that all the items are in range 0 to 0.75 support count and if the min-support exceeds 76% then none of the items will be considered as frequent
set item and no rules will generated after on. so, to skip this case we have considered the support values which are less than 76 as minimum support. According to algorithm we can conclude that higher the support count will give no rules.

Manual work to calculate sample datasets

For Sample report, we took transaction 193,194,195,196

So basically, it will a calculate the frequency of all items in item list of 1 items 2 items ,3 items ,,,,,,, n items until it will get reach it support level, mean frequency of items should be satisfying the support level formulae which was given earlier

Here as we took total transactions as 4,, item frequency – for particular item

- **Support(i(1st item)) =** Item’s frequency/total number of transactions

TID's	Transactions									

T193	paper towels	cheeses	vegetables	ketchup	juice	sandwich loaves	ice cream	dinner rolls	soap	beef	toilet paper
T194	pork	spaghetti sauce	milk	sandwich bags	ketchup	sandwich loaves	pasta	bagels	sugar		
T195	soap	laundry detergent	coffee/tea	tortillas	ketchup	sandwich loaves	pasta				
T196	flour	soda	milk	all-purpose	fruits	sandwich loaves	ice cream	dinner rolls	beef		

Frequency – 1 item list

```
ITEMS. ----------- frequency
(' milk'). ------      2
(' beef') ------      2
(' sandwich bags') ------      2
(' soap') ------      2
(' pasta') ------      2
(' ketchup') ------      3
(' ice cream') ------      2
(' dinner rolls') ------      2
(' sandwich loaves') ------      4
```

Frequency – 2 item list

```
ITEMS. ----------- frequency
(' beef', ' ice cream') ------      2
(' sandwich loaves', ' ice cream') ------      2
(' ketchup', ' soap') ------      2
(' beef', ' dinner rolls') ------      2
(' ketchup', ' sandwich loaves') ------      3
(' sandwich loaves', ' pasta') ------      2
(' sandwich loaves', ' soap') ------      2
(' dinner rolls', ' ice cream') ------      2
(' sandwich bags', ' sandwich loaves') ------      2
(' beef', ' sandwich loaves') ------      2
(' sandwich loaves', ' milk') ------      2
(' dinner rolls', ' sandwich loaves') ------      2
```

Frequency – 3 item list

```
ITEMS. ----------- frequency
(' ketchup', ' sandwich loaves', ' soap') ------      2
(' sandwich loaves', ' beef', ' ice cream') ------      2
(' beef', ' dinner rolls', ' sandwich loaves') ------      2
(' beef', ' dinner rolls', ' ice cream') ------      2
(' dinner rolls', ' sandwich loaves', ' ice cream') ------      2
```

Frequency – 4 item list

```
ITEMS. ----------- frequency
(' beef', ' dinner rolls', ' sandwich loaves', ' ice cream') ------      2
```

Case1:

For case1 to evaluate results we use minimum support = 0.3 & minimum confidence = 0.5

So basically, it will a calculate the frequency of all items in item list of 1 items 2 items ,3 items ,,,,,,, n items until it will get reach it support level

So, support for each of the item in item list can be calculated as in frequency divides by total no of transactions

So for example for milk,,, its frequency is 2 as in T194 , T196 and total transactions is 4 so support will be 2/4 = .5 which is 50 %

Similarly sandwich `loaves, ice cream` its frequency is 2 appears in T193,T196 and total transactions is 4 so support will be 2/4 = .5 which is 50 %

Done for all other itemset

```
Itemset ----------- Support in percentages
 milk, 50
  dinner rolls, 50
  beef, 50
  sandwich bags, 50
  soap, 50
  pasta, 50
  ice cream, 50
  beef,  sandwich loaves , 50
  sandwich loaves,  soap , 50
  sandwich loaves,  dinner rolls , 50
  sandwich loaves,  milk , 50
 ketchup,  sandwich bags , 50
  beef,  dinner rolls , 50
  ice cream,  sandwich loaves , 50
 ketchup,  soap , 50
  ice cream,  beef , 50
  sandwich loaves,  pasta , 50
 ketchup,  pasta , 50
  ice cream,  dinner rolls , 50
  sandwich loaves,  sandwich bags , 50
  ice cream,  beef,  dinner rolls , 50
  beef,  sandwich loaves,  dinner rolls , 50
  ice cream,  beef,  sandwich loaves , 50
 ketchup,  sandwich loaves,  soap , 50
  ice cream,  sandwich loaves,  dinner rolls , 50
 ketchup,  sandwich loaves,  sandwich bags , 50
 ketchup,  sandwich loaves,  pasta , 50
  ice cream,  sandwich loaves,  dinner rolls,  beef , 50
 ketchup, 75
 ketchup,  sandwich loaves , 75
  sandwich loaves, 100
```

For the given support it calculates and check the minimum confidence after getting its non-empty subsets are calculated which have confidence intervals

For suppose we can only start with 2 item lists as it needs to be calculated with more than 1 no empty sets so

For suppose for item list. (' beef', ' ice cream') ------ 2
It has frequency 2, support – 50 ------------Non empty sets - { (beef), (ice cream) }

For suppose for item list. (' ketchup', ' sandwich loaves', ' soap') ------ 2
It has frequency 2, support – 50 -------Non empty sets - { (ketchup), (sandwich loaves), (soap) ,(ketchup, sandwich loaves), (soap, ketchup),(sandwich loaves, soap)}

Similarly, non-empty list will be defined for every other item list

Now that we have non-empty item lists, we can get some rule which will help to get confidence value late r the confidence value it matched with our min confidence value and be verified if rule is pass or fail

For suppose we can take up the rules for (' beef', ' ice cream')

From rules we got 2 sets which are non-empty

So
Rule ------ beef => ice cream
Support (beef, ice-cream) 50
Confidence = support of itemset/ support of nonempty itemset
Confidence = Support (beef, ice-cream) / Support(beef)
Support(beef) = 50
Now confidence = 50 /50 = 1
As confidence is greater than our confidence level rule passed

Similarly
Rule ------ ice cream => beef
Support (beef, ice-cream) 50
Support (ice cream) = 50
Now confidence = 50 /50 = 1
As confidence is greater than our confidence level rule passed

Both rules passed > or =

Now similarly rules which will be passed are mentioned below along with confidence level

RULES--------------- Confidence
sandwich loaves, ==> beef, 0.5
sandwich loaves, ==> soap, 0.5
sandwich loaves, ==> dinner rolls, 0.5
sandwich loaves, ==> milk, 0.5
sandwich loaves, ==> ice cream, 0.5
sandwich loaves, ==> pasta, 0.5
sandwich loaves, ==> sandwich bags, 0.5

```
 sandwich loaves, ==>  beef,  dinner rolls , 0.5
 sandwich loaves, ==>  ice cream,  beef , 0.5
 sandwich loaves, ==> ketchup,  soap , 0.5
 sandwich loaves, ==>  ice cream,  dinner rolls , 0.5
 sandwich loaves, ==> ketchup,  sandwich bags , 0.5
 sandwich loaves, ==> ketchup,  pasta , 0.5
 sandwich loaves, ==>  ice cream,  beef,  dinner rolls , 0.5
ketchup, ==>  sandwich bags, 0.667
ketchup, ==>  soap, 0.667
ketchup, ==>  pasta, 0.667
ketchup, ==>  sandwich loaves,  soap , 0.667
ketchup,  sandwich loaves ==>  soap, 0.667
ketchup, ==>  sandwich loaves,  sandwich bags , 0.667
ketchup,  sandwich loaves ==>  sandwich bags, 0.667
ketchup, ==>  sandwich loaves,  pasta , 0.667
ketchup,  sandwich loaves ==>  pasta, 0.667
 sandwich loaves, ==> ketchup, 0.750
 beef, ==>  sandwich loaves, 1
 soap, ==>  sandwich loaves, 1
 dinner rolls, ==>  sandwich loaves, 1
 milk, ==>  sandwich loaves, 1
 sandwich bags, ==> ketchup, 1
 beef, ==>  dinner rolls, 1
 dinner rolls, ==>  beef, 1
ketchup, ==>  sandwich loaves, 1
 ice cream, ==>  sandwich loaves, 1
 soap, ==> ketchup, 1
 ice cream, ==>  beef, 1
 beef, ==>  ice cream, 1
 pasta, ==>  sandwich loaves, 1
 pasta, ==> ketchup, 1
 ice cream, ==>  dinner rolls, 1
 dinner rolls, ==>  ice cream, 1
 sandwich bags, ==>  sandwich loaves, 1
 ice cream, ==>  beef,  dinner rolls , 1
 beef, ==>  ice cream,  dinner rolls , 1
 dinner rolls, ==>  ice cream,  beef , 1
 ice cream,  beef ==>  dinner rolls, 1
 ice cream,  dinner rolls ==>  beef, 1
 beef,  dinner rolls ==>  ice cream, 1
 beef, ==>  sandwich loaves,  dinner rolls , 1
 dinner rolls, ==>  beef,  sandwich loaves , 1
 beef,  sandwich loaves ==>  dinner rolls, 1
 beef,  dinner rolls ==>  sandwich loaves, 1
 sandwich loaves,  dinner rolls ==>  beef, 1
 ice cream, ==>  beef,  sandwich loaves , 1
 beef, ==>  ice cream,  sandwich loaves , 1
 ice cream,  beef ==>  sandwich loaves, 1
 ice cream,  sandwich loaves ==>  beef, 1
 beef,  sandwich loaves ==>  ice cream, 1
 soap, ==> ketchup,  sandwich loaves , 1
```

```
ketchup,  soap ==>  sandwich loaves, 1
 sandwich loaves,  soap ==> ketchup, 1
 ice cream, ==>  sandwich loaves,  dinner rolls , 1
 dinner rolls, ==>  ice cream,  sandwich loaves , 1
 ice cream,  sandwich loaves ==>  dinner rolls, 1
 ice cream,  dinner rolls ==>  sandwich loaves, 1
 sandwich loaves,  dinner rolls ==>  ice cream, 1
 sandwich bags, ==> ketchup,  sandwich loaves , 1
ketchup,  sandwich bags ==>  sandwich loaves, 1
 sandwich loaves,  sandwich bags ==> ketchup, 1
 pasta, ==> ketchup,  sandwich loaves , 1
ketchup,  pasta ==>  sandwich loaves, 1
 sandwich loaves,  pasta ==> ketchup, 1
 ice cream, ==>  beef,  sandwich loaves,  dinner rolls , 1
 dinner rolls, ==>  ice cream,  beef,  sandwich loaves , 1
 beef, ==>  ice cream,  sandwich loaves,  dinner rolls , 1
 ice cream,  sandwich loaves ==>  beef,  dinner rolls , 1
 ice cream,  dinner rolls ==>  beef,  sandwich loaves , 1
 ice cream,  beef ==>  sandwich loaves,  dinner rolls , 1
 sandwich loaves,  dinner rolls ==>  ice cream,  beef , 1
 beef,  sandwich loaves ==>  ice cream,  dinner rolls , 1
 beef,  dinner rolls ==>  ice cream,  sandwich loaves , 1
 ice cream,  sandwich loaves,  dinner rolls ==>  beef, 1
 ice cream,  beef,  sandwich loaves ==>  dinner rolls, 1
 ice cream,  beef,  dinner rolls ==>  sandwich loaves, 1
 beef,  sandwich loaves,  dinner rolls ==>  ice cream, 1
```

here we can see we got lot of values in both confidence and support now in case 2 we try to increase support and confidence to .75 and check the rules which can be passed

Case2:

For case2 to evaluate results we use minimum support = 0.75 & minimum confidence = 0.75

So basically, it will a calculate the frequency of all items in item list of 1 items 2 items ,3 items ,,,,,,, n items until it will get reach it support level

So, support for each of the item in item list can be calculated as in frequency divides by total no of transactions

So for example for ketchup,,, its frequency is 3 as in T193 ,T194 , T195 and total transactions is 4 so support will be
3/4 = .75 which is 75 %

Similarly sandwich loaves, ketchup its frequency is 3 appears in T193 ,T194 , T195 and total transactions is 4 so support will be 3/4 = .75 which is 75 %

Done for all other itemset

```
ITEMS------------Support
```

```
ketchup, 75
ketchup, sandwich loaves, 75
sandwich loaves, 100
```

For the given support it calculates and check the minimum confidence after getting its non-empty subsets are calculated which have confidence intervals

For suppose we can only start with 2 item lists as it needs to be calculated with more than 1 no empty sets so

For suppose for item list. (' sandwich loaves, ' ketchup ') ------
2
It has frequency 3, support – 75 ------------Non empty sets - { (sandwich loaves), (ketchup) }

Similarly, non-empty list will be defined for every other item list

Now that we have non-empty item lists, we can get some rule which will help to get confidence value later the confidence value it matched with our min confidence value and be verified if rule is pass or fail

For suppose we can take up the rules for (' sandwich loaves, ' ice cream')

From rules we got 2 sets which are non-empty

So
Rule ------ sandwich loaves => ketchup
Support (sandwich loaves, ketchup)75
Confidence = support of itemset/ support of nonempty itemset
Confidence = Support (sandwich loaves, ketchup) / Support(sandwich loaves)
Support (sandwich loaves) = 100
Now confidence = 75 /100 = .75
As confidence is greater than our confidence level rule passed

Similarly
Rule ------ ketchup => sandwich loaves
Support (sandwich loaves, ketchup) 75
Support (ketchup) = 75
Now confidence = 75 /75 = 1
As confidence is greater than our confidence level rule passed

Both rules passed > or = 0.75 both rules passed

Now similarly rules which will be passed are mentioned below along with confidence level

```
RULES------------Confidence
sandwich loaves==> ketchup, 0.75
ketchup==> sandwich loaves, 1
```

as of now we can see still the rules are passed our min confidence ill try to increase the support further more to check if any item is being repeated in almost all transactions

Case3:

For case3 to evaluate results we use minimum support = 0.76 & minimum confidence = 0.1

Here we only get item with support level more that 76 and min confidence greater than 10 %

But as we get only 1 item which satisfies this support level condition where support level is greater than 75 item should be in every transaction as we have 4 transactions

I can only see one item sandwich loaves- which is present in all 4 transactions T193, T194,T195,T196

```
ITEMS-----------------
sandwich loaves, 100
```

```
RULES-----------------
```

As we have no non – empty sets we cannot have any rules for the following conditions