Part 1

Question 1

Question 2 Advantage of using the Apriori algorithm in comparison with computing the support of every subset of an itemset in order to find the frequent itemsets in a transaction dataset? (0.5 marks out of 2)

Instead of computing support count for every subset of transactional itemset, the Apriori algorithm reduces the search space by eliminating those subsets that fall below the pre-defined support count threshold. This greatly improves the efficiency of the frequent itemset mining process.

Question 3 Given the set of frequent 3-itemsets $\{a, b, c\}$, $\{a, b, d\}$ and every frequent 2-itemset for a dataset of six transactions $\{a, c\}$, $\{b, d\}$ (with size of 6).

1) Generate the set of frequent 3-itemsets. Hint: $\{a, c\}$ and every frequent 2-itemset for a dataset of six transactions $\{a, c\}$, $\{b, d\}$ (with size of 6)

2) Generate the set of frequent 4-itemsets. Hint: $\{a, c\}$ and every frequent 2-itemset for a dataset of six transactions $\{a, c\}$, $\{b, d\}$ (with size of 6)

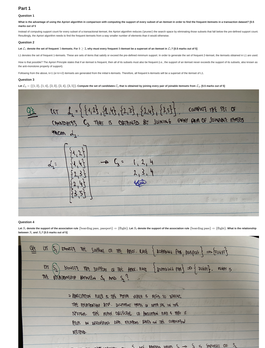
3) Given the set of frequent 3-itemsets, prove that a 4-itemset is frequent. Hint: all of its subsets must also be frequent. Hint: $\{a, c\}$ and every frequent 2-itemset for a dataset of six transactions $\{a, c\}$, $\{b, d\}$ (with size of 6)

4) Given the set of frequent 3-itemsets, prove that a 4-itemset is frequent. Hint: all of its subsets must also be frequent. Hint: $\{a, c\}$ and every frequent 2-itemset for a dataset of six transactions $\{a, c\}$, $\{b, d\}$ (with size of 6)

Following from the above, $\{a, c\}$ $\{b, d\}$ itemsets are generated from the initial 3-itemsets. Therefore, all frequent 4-itemsets will be a superset of the itemset $\{a, c\}$.

Question 5

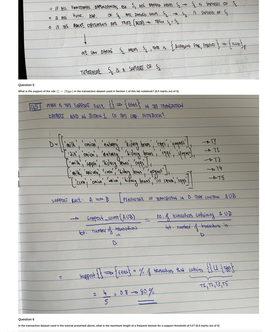
Let $L_1 = \{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$. Compute the set of candidates C_2 that is obtained by joining every pair of possible itemsets from L_1. (0.5 marks out of 3)



Question 4

Let S. Denote the support of the association rule $\{\text{Toothbrush}, \text{pen}, \text{pencil}\} \rightarrow \{\text{Apple}\}$ and $\{\text{Toothbrush}, \text{pen}, \text{pencil}\} \rightarrow \{\text{Pear}\}$. What is the relationship between S and S'?

What is the relationship between S and S'?



Question 5

What is the value of the lift $\{\{\text{Tea}\}\} \rightarrow \{\text{Pen}\}$ in the transaction dataset used in Section 2 of this lab notebook? (0.5 marks out of 2)

Question 6

In the transaction dataset used in the tutorial presented above, what is the maximum length of a frequent itemset for a support threshold of 0.2? (0.5 marks out of 2)

```
jup install mlxtend

Requirement already satisfied: mlxtend in /usr/local/lib/python3.8/dist-packages (0.19.0)
Requirement already satisfied: numpy>=1.16.2 in /usr/local/lib/python3.8/dist-packages (from mlxtend) (1.19.5)
Requirement already satisfied: scipy>=1.2.1 in /usr/local/lib/python3.8/dist-packages (from mlxtend) (1.4.1)
Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.8/dist-packages (from mlxtend) (1.1.5)
Requirement already satisfied: scikit-learn>=0.20.3 in /usr/local/lib/python3.8/dist-packages (from mlxtend) (1.0.2)
Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from mlxtend) (3.2.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from mlxtend) (57.4.0)
Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.8/dist-packages (from mlxtend) (1.1.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3.0.0 mlxtend) (1.4.4)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3.0.0 mlxtend) (0.11.0)
Requirement already satisfied: pyparsing>=2.4.7 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3.0.0 mlxtend) (3.0.9)
Requirement already satisfied: dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3.0.0 mlxtend) (2.8.2)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.8/dist-packages (from pandas>=0.24.2 mlxtend) (2022.6)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from scikit-learn>=0.20.3 mlxtend) (3.1.0)
Requirement already satisfied: typing_extensions in /usr/local/lib/python3.8/dist-packages (from kiwisolver>=1.0.1 mlxtend) (4.4.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from dateutil>=2.1 mlxtend) (1.15.0)
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.8/dist-packages (from typing_extensions in /usr/local/lib/python3.8/dist-packages) (3.10.0)
Mkdir: cmd: mkdir -p /root/.local/share/jupyter/kernels/python3
Installed kernelspec python3 in /root/.local/share/jupyter/kernels/python3
```

```
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
import pandas as pd
from mlxtend.frequent_patterns import association_rules

dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
df
```

	Apple	Dill	Eggs	Ice cream	Kidney Beans	Milk	Nutmeg	Onion	Unicorn	Corn	Yogurt
0	False	True	True	False	True	True	True	True	False	False	True
1	False	True	True	False	True	False	True	True	False	False	True
2	True	False	True	False	False	True	False	False	False	False	False
3	False	False	False	False	True	True	False	False	True	True	True
4	False	False	True	True	True	False	False	True	False	True	False

```
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)
frequent_itemsets
```

	support	itemsets
0	0.8	(Eggs)
1	0.6	(Kidney Beans)
2	0.8	(Milk)
3	0.8	(Onion)
4	0.6	(Yogurt)
5	0.6	(Eggs, Kidney Beans)
6	0.6	(Eggs, Onion)
7	0.6	(Eggs, Yogurt)
8	0.6	(Kidney Beans, Onion)
9	0.6	(Kidney Beans, Yogurt)
10	0.6	(Onion, Eggs, Kidney Beans)
11	0.6	(Eggs, Kidney Beans, Yogurt)
12	0.6	(Onion, Kidney Beans, Yogurt)
13	0.6	(Onion, Eggs, Kidney Beans, Yogurt)

Still error 1 solved

```
frequent_itemsets[ frequent_itemsets['itemsets'] == {'Eggs', 'Kidney Beans'} ]
```

	support	itemsets
5	0.6	(Eggs, Kidney Beans)

```
# finding the maximum value
max_length = frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets[ max_length == max(max_length) ]
```

	support	itemsets
13	0.6	(Onion, Eggs, Kidney Beans, Yogurt)

Still error 2 solved

```
frequent_itemsets[ frequent_itemsets['itemsets'] == {'Eggs', 'Kidney Beans', 'Yogurt'} ]
```

	support	itemsets
11	0.6	(Eggs, Kidney Beans, Yogurt)

```
# finding the maximum value
max_length = frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets[ max_length == max(max_length) ]
```

	support	itemsets
13	0.6	(Onion, Eggs, Kidney Beans, Yogurt)

Still error 3 solved

```
association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Eggs)	(Kidney Beans)	0.8	0.6	0.6	0.75	1.25	0.120000	1.6
1	(Eggs)	(Onion)	0.8	0.8	0.6	0.75	0.9375	-0.040000	0.8
2	(Onion)	(Eggs)	0.8	0.8	0.6	0.75	0.9375	-0.040000	0.8
3	(Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
4	(Onion)	(Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
5	(Eggs)	(Kidney Beans, Onion)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
6	(Kidney Beans)	(Eggs, Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
7	(Onion)	(Eggs, Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
8	(Eggs, Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
9	(Eggs, Onion)	(Kidney Beans)	0.6	0.6	0.6	1.0	1.666667	0.240000	inf
10	(Kidney Beans, Onion)	(Eggs)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf

```
# initialize a variable
confidence = 0.7
# calculate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=confidence)
# display the rules
rules[rules['antecedents'] == {'Onion'}]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
4	(Onion)	(Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
7	(Onion)	(Eggs, Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6

```
# initialize a variable
confidence = 0.8
# calculate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=confidence)
# display the rules
rules[rules['antecedents'] == {'Onion'}]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
3	(Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
8	(Eggs, Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf

```
# initialize a variable
confidence = 0.9
# calculate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=confidence)
# display the rules
rules[rules['antecedents'] == {'Onion'}]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
3	(Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
8	(Eggs, Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf

The frequently observed lift consequent is 1.25000 and is Kidney Beans, Eggs, Onion, Yogurt.

Question 7

Implement a function that computes the Kulczynski measure of two itemsets. Let X and Y be two transactions. Hint: $\{\text{Milk}, \text{Onion}, \text{Eggs}\}$ in the transaction dataset used in this lab notebook. (1 mark out of 3)

Implement a function that computes the Kulczynski measure of two itemsets. Let X and Y be two transactions. Hint: $\{\text{Milk}, \text{Onion}, \text{Eggs}\}$ in the transaction dataset used in this lab notebook. (1 mark out of 3)

```
string_list = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
display(string_list)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Eggs)	(Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
1	(Eggs)	(Onion)	0.8	0.8	0.6	0.75	0.9375	-0.040000	0.8
2	(Onion)	(Eggs)	0.8	0.8	0.6	0.75	0.9375	-0.040000	0.8
3	(Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
4	(Onion)	(Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
5	(Eggs)	(Kidney Beans, Onion)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
6	(Kidney Beans)	(Eggs, Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
7	(Onion)	(Eggs, Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
8	(Eggs, Kidney Beans)	(Onion)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf
9	(Eggs, Onion)	(Kidney Beans)	0.6	0.6	0.6	1.0	1.666667	0.240000	inf
10	(Kidney Beans, Onion)	(Eggs)	0.6	0.8	0.6	1.0	1.250000	0.120000	inf

```
# initialize a variable
confidence = 0.7
# calculate association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=confidence)
# display the rules
rules[rules['antecedents'] == {'Onion'}]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
4	(Onion)	(Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6
7	(Onion)	(Eggs, Kidney Beans)	0.8	0.6	0.6	0.75	1.250000	0.120000	1.6

The frequently observed lift consequent is 1.25000 and is Kidney Beans, Eggs, Onion.

Question 8

Implement a function that computes the Kulczynski measure of two itemsets. Let X and Y be two transactions. Hint: $\{\text{Milk}, \text{Onion}, \text{Eggs}\}$ in the transaction dataset used in this lab notebook. (1 mark out of 3)

Implement a function that computes the Kulczynski measure of two itemsets. Let X and Y be two transactions. Hint: $\{\text{Milk}, \text{Onion}, \text{Eggs}\}$ in the transaction dataset used in this lab notebook. (1 mark out of 3)

```
support = frequent_itemsets[frequent_itemsets['itemsets'] == {'Onion'}]
support_X = support['support'].iloc[0]
support = frequent_itemsets[frequent_itemsets['itemsets'] == {'Eggs', 'Kidney Beans'}]
support_Y = support['support'].iloc[0]
support = frequent_itemsets[frequent_itemsets['itemsets'] == {'Onion', 'Eggs', 'Kidney Beans'}]
support_XY = support['support'].iloc[0]
support_X = 0.8
support_Y = 0.6
support_XY = 0.6
kulczynski = 1/2 * (support_XY/support_X) + (support_XY/support_Y)
print("The Kulczynski measure of Onion and Eggs, Kidney Beans is: ", kulczynski)
```

```
Mount of Column 1:     support  itemsets
0  0.8  (Eggs)
1  0.6  (Kidney Beans)
2  0.8  (Milk)
3  0.8  (Onion)
4  0.6  (Yogurt)
Mount of Column 2:     support  itemsets
0  0.6  (Eggs, Kidney Beans)
1  0.6  (Eggs, Onion)
2  0.6  (Eggs, Yogurt)
3  0.6  (Kidney Beans, Onion)
4  0.6  (Kidney Beans, Yogurt)
5  0.6  (Onion, Eggs, Kidney Beans)
6  0.6  (Eggs, Kidney Beans, Yogurt)
7  0.6  (Onion, Kidney Beans, Yogurt)
8  0.6  (Onion, Eggs, Kidney Beans, Yogurt)
The Kulczynski measure of Onion and Eggs, Kidney Beans is:  0.9375
```

Part 2

Question 1

Use the Iris dataset and explore every pair of your suggested attributes from the Week 1 lab, and that you are asked to detect outliers from this dataset. Additional information on the dataset attributes can be found here. Explain one possible outlier detection method that you could apply for detecting outliers for the particular dataset. Explain what is a definition of an outlier for your suggested approach (from this particular dataset), and justify why would you choose this particular method for outlier detection. (1 mark out of 5)

Question 2

The monthly rainfall in the Cooma township of Tuross Heights in 2018 has the following amount of precipitation (measured in mm, values from January-December: 80.60, 22.00, 30.70, 25.00, 12.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00). According to the Cooma township, the mean and standard deviation of the rainfall in Cooma township are the maximum likelihood estimates (1 mark out of 5)

$$\text{MLE}(\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i = 22.57$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})^2 = 46.9155 \quad \Rightarrow \sigma = \sqrt{46.9155} = 6.8516$$

Ignoring bias or Bessel's correction

$\{0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 12.00, 22.00, 25.00, 30.70, 80.60\}$