(a) A drawback of using confidence is that it ignores Pr(B). Why is this a drawback? Explain why lift and conviction do not suffer from this drawback.

Solution:
The formula of confidence is conf(A → B) = Pr(B|A). there is a chance that an association has a high confidence but it is not interesting.
For example, X->milk has a high confidence may just because milk is purchases very often. In fact, milk is independent with X.
The formula of lift is lift(A → B) = conf(A → B)/ S(B) and the formula of conviction is conv(A → B) = 1 − S(B)/ 1 − conf(A → B).

They all consider the support of B and could not suffer from the drawback that an association with a high confidence may not be interesting.

(b) A measure is symmetrical if measure(A → B) = measure(B → A). Which of the measures presented here are symmetrical? For each measure, please provide either a proof that the measure is symmetrical, or a counterexample that shows the measure is not symmetrical.

Solution:

1. the confidence is asymmetrical.
   Conf(A->B)=P(AB)/P(A)
   Conf(B->A)=P(AB)/P(B)
   Conf(A->B)≠Conf(B->A) when P(A)≠P(B), which is very common.
2. The lift is symmetrical.
   lift(A → B) = conf(A → B)/ S(B)= P(AB)/P(A)P(B)
   lift(B → A) = conf(B → A)/ S(A)= P(AB)/P(A)P(B)
3. The conviction is asymmetrical.

   conv(A → B) = 1 − S(B)/ 1 − conf(A → B) . it is the probability that A appears without B

   conv(B→ A) = 1 − S(A)/ 1 − conf(B→ A). it is the probability that B appears without A

   for example:

$B_1 = \{m, c, b\}$	$B_2 = \{m, p, j\}$
$B_3 = \{m, b\}$	$B_4 = \{c, j\}$
$B_5 = \{m, p, b\}$	$B_6 = \{m, c, b, j\}$
$B_7 = \{c, b, j\}$	$B_8 = \{b, c\}$

   Conv({m,b}->c)=3/4
   Conv(c->{m,b})=5/6

(c) Perfect implications are rules that hold 100% of the time (or equivalently, the associated conditional probability is 1). A measure is desirable if it reaches its maximum achievable value for all perfect implications. This makes it easy to identify the best rules. Which of the above measures have this property? You may ignore 0/0 but not other infinity cases. Also you may find it easy to explain by an example.

Solution:
1.Lift has this property.
When a rule holds 100% such as A->B, then P(AB)=P(A)
Lift(A->B)=P(AB)/P(A)P(B)=1/P(B)=MAX(Lift(A->B))

2.conviction has this property.
When a rule holds 100% such as A->B, then P(AB)=P(A)

$conv(A \rightarrow B) = 1 - S(B)/1 - conf(A \rightarrow B) = +\infty = MAX(Conv(A->B))$

(d) Identify pairs of items (X, Y) such that the support of {X, Y } is at least 100. For all such pairs, compute the confidence scores of the corresponding association rules: X ⇒ Y , Y ⇒ X.Sort the rules in decreasing order of confidence scores and list the top 5 rules in the writeup.Break ties, if any, by lexicographically increasing order on the left hand side of the rule.
(You need not use Spark for parts d and e)

(e) Identify item triples (X, Y, Z) such that the support of {X, Y, Z} is at least 100. For all such triples, compute the confidence scores of the corresponding association rules: (X, Y) ⇒ Z,(X, Z) ⇒ Y , (Y, Z) ⇒ X. Sort the rules in decreasing order of confidence scores and list the top 5 rules in the writeup. Order the left-hand-side pair lexicographically and break ties, if any, by lexicographical order of the first then the second item in the pair.

Solution:
1. codes:

```
import pandas as pd
from apyori import apriori
#导入和处理数据
dataset = pd.read_csv(r'D:\冲鸭！\上财研究生事务相关\上课相关\研一下\人工智能\作业\Assignment1\Assignment1\data\browsing.txt',header=None)
res = dataset[0].apply(lambda x:x.split(" ")).values
data= []
for list_row in res:
    new_list = list(set([i for i in list_row if i != ""]))
    data.append(new_list)

from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
```

```
import pandas as pd
te = TransactionEncoder()
#编码
te_ary = te.fit(data).transform(data)     #类似 onehot 编码，所有的商品都是特征，买了的样本对应 1，没买的样本对应 0
df = pd.DataFrame(te_ary, columns=te.columns_)
freq=apriori(df,min_support=100/31101, use_colnames=True,max_len=3)
#导入关联规则包
from mlxtend.frequent_patterns import association_rules
#计算关联规则
result = association_rules(freq, metric="confidence", min_threshold=0.4)
```

2. for (x,y), the top 5 are

antecedents	consequents	anteceden	consequen	support	confidence	lift	leverage	conviction
frozenset({'DAI93865'})	frozenset({'FRO40251'})	0.006688	0.124787	0.006688	1	8.013656	0.005853	inf
frozenset({'GRO85051'})	frozenset({'FRO40251'})	0.039034	0.124787	0.039002	0.999176	8.007055	0.034131	1062.509
frozenset({'GRO38636'})	frozenset({'FRO40251'})	0.00344	0.124787	0.003408	0.990654	7.938762	0.002979	93.64779
frozenset({'ELE12951'})	frozenset({'FRO40251'})	0.003408	0.124787	0.003376	0.990566	7.938056	0.002951	92.77258
frozenset({'DAI88079'})	frozenset({'FRO40251'})	0.014533	0.124787	0.01434	0.986726	7.90728	0.012527	65.93271

3. for(X,Y,Z), the top 5 are

antecedents	consequents	anteceden	consequen	support	confidence	lift	leverage	conviction
frozenset({'DAI23334', 'ELE92920'})	frozenset({'DAI62779'})	0.004598	0.214366	0.004598	1	4.664917	0.003612	inf
frozenset({'DAI31081', 'GRO85051'})	frozenset({'FRO40251'})	0.00328	0.124787	0.00328	1	8.013656	0.00287	inf
frozenset({'DAI55911', 'GRO85051'})	frozenset({'FRO40251'})	0.004276	0.124787	0.004276	1	8.013656	0.003743	inf
frozenset({'DAI62779', 'DAI88079'})	frozenset({'FRO40251'})	0.003762	0.124787	0.003762	1	8.013656	0.003292	inf
frozenset({'GRO85051', 'DAI75645'})	frozenset({'FRO40251'})	0.012701	0.124787	0.012701	1	8.013656	0.011116	inf