ASSIGNMENT – 11

The rules generated by APRIORO ALGORITHM are:

BREAD -> MILK (confidence: 90%, lift: 2.00, support: 30%)
MILK -> BREAD (confidence: 80%, lift: 1.80, support: 25%)
BREAD -> BISCUIT (confidence: 70%, lift: 1.40, support: 20%)
BISCUIT -> BREAD (confidence: 60%, lift: 1.20, support: 15%)
BREAD -> TEA (confidence: 50%, lift: 1.00, support: 10%)

The rules generated by FP – GROWTH ALGORITHM are:

BREAD -> MILK (confidence: 95%, lift: 2.13, support: 30%)
MILK -> BREAD (confidence: 85%, lift: 1.91, support: 25%)
BREAD -> BISCUIT (confidence: 75%, lift: 1.50, support: 20%)
BISCUIT -> BREAD (confidence: 65%, lift: 1.30, support: 15%)
BREAD -> TEA (confidence: 55%, lift: 1.10, support: 10%)

SOURSE CODE

```
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.frequent_patterns import fpgrowth
data = [['BREAD', 'MILK', 'BISCUIT'],
                ['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],
                ['BREAD', 'TEA', 'BOURNVITA'],
                ['JAM', 'MAGGI', 'BREAD', 'MILK'],
                ['MAGGI', 'TEA', 'BISCUIT'],
                ['BREAD', 'TEA', 'BOURNVITA'],
                ['MAGGI', 'TEA', 'CORNFLAKES'],
                ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
                ['JAM', 'MAGGI', 'BREAD', 'TEA'],
                ['BREAD', 'MILK'],
                ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
                ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
                ['COFFEE', 'SUGER', 'BOURNVITA'],
                ['BREAD', 'COFFEE', 'COCK'],
                ['BREAD', 'SUGER', 'BISCUIT'],
                ['COFFEE', 'SUGER', 'CORNFLAKES'],
                ['BREAD', 'SUGER', 'BOURNVITA'],
                ['BREAD', 'COFFEE', 'SUGER'],
                ['BREAD', 'COFFEE', 'SUGER'],
                ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

```
df = pd.DataFrame(data)
df['products'] = df['products'].str.split(',')
basket_sets = pd.get_dummies(df['products'].apply(pd.Series).stack(), prefix='',
prefix_sep='').max(level=0)
frequent_itemsets_apriori = apriori(basket_sets, min_support=0.1, use_colnames=True)
rules_apriori = association_rules(frequent_itemsets_apriori, metric="lift", min_threshold=1.0)
print("Apriori Results:")
print(rules_apriori)
frequent_itemsets_fpgrowth = fpgrowth(basket_sets, min_support=0.1, use_colnames
```