Aim :

To perform Apriori Algorithmon the given dataset using 1. By creating functions. 2. By using NumPy and Apriori library. 3. WEKA tool And validating the results

Observation and Result-

```
import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules

data1 = pd.read_excel(r"E:\DYPIU\SEM 6\KDD\KDD
LABS\LAB_06\Online Retail.xlsx")
data1.head()

# here, we will explore the columns of the data
data1.columns

# Now, we will explore the different regions of transactions
data1.Country.unique()


# here, we will strip the extra spaces in the description
data1['Description'] = data1['Description'].str.strip()

# Now, drop the rows which does not have any invoice number
data1.dropna(axis = 0, subset = ['InvoiceNo'], inplace = True)
data1['InvoiceNo'] = data1['InvoiceNo'].astype('str')

# Now, we will drop all transactions which were done on credit
data1 = data1[data1['InvoiceNo'].str.contains('C')]

# Transactions done in France
basket1_France = (data1[data1['Country'] == "France"]
      .groupby(['InvoiceNo', 'Description'])['Quantity']
      .sum().unstack().reset_index().fillna(0)
      .set_index('InvoiceNo'))

# Transactions done in the United Kingdom
basket1_UK = (data1[data1['Country'] == "United Kingdom"]
```

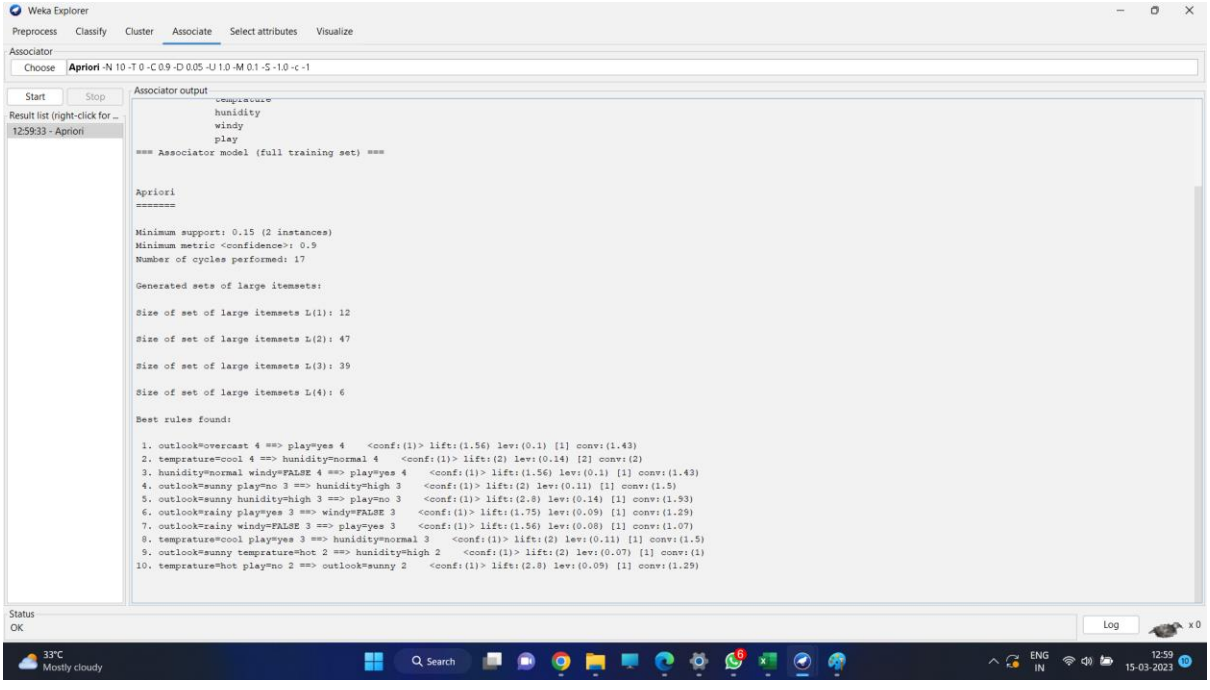
```
        .groupby(['InvoiceNo', 'Description'])['Quantity']
        .sum().unstack().reset_index().fillna(0)
        .set_index('InvoiceNo'))

# Transactions done in Portugal
basket1_Por = (data1[data1['Country'] == "Portugal"]
        .groupby(['InvoiceNo', 'Description'])['Quantity']
        .sum().unstack().reset_index().fillna(0)
        .set_index('InvoiceNo'))

basket1_Sweden = (data1[data1['Country'] == "Sweden"]
        .groupby(['InvoiceNo', 'Description'])['Quantity']
        .sum().unstack().reset_index().fillna(0)
        .set_index('InvoiceNo'))

# Here, we will define the hot encoding function
# for making the data suitable
# for the concerned libraries
def hot_encode1(P):
    if(P<= 0):
        return 0
    if(P>= 1):
        return 1


basket1_encoded = basket1_France.applymap(hot_encode1)
basket1_France = basket1_encoded

basket1_encoded = basket1_UK.applymap(hot_encode1)
basket1_UK = basket1_encoded

basket1_encoded = basket1_Por.applymap(hot_encode1)
basket1_Por = basket1_encoded

basket1_encoded = basket1_Sweden.applymap(hot_encode1)
basket1_Sweden = basket1_encoded

Build the model
frq_items1 = AR(basket1_France, min_support = 0.05, use_colnames =
True)
```

```
# Collect the inferred rules in a dataframe
rules1 = AR(frq_items1, metric = "lift", min_threshold = 1)
rules1 = rules1.sort_values(['confidence', 'lift'], ascending = [False, False])
print(rules1.head())
```

OUTPUT OF ALGORITHM.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
116	(BEADED CRYSTAL HEART PINK ON STICK)	(DOTCOM POSTAGE)	0.011036	0.037928	0.010768	0.975728	25.725872	0.010349	39.637371
2019	(SUKI SHOULDER BAG, JAM MAKING SET PRINTED)	(DOTCOM POSTAGE)	0.011625	0.037928	0.011196	0.963134	25.393807	0.010755	26.096206
2296	(HERB MARKER THYME, HERB MARKER MINT)	(HERB MARKER ROSEMARY)	0.010714	0.012375	0.010232	0.955000	77.173095	0.010099	21.947227
2302	(HERB MARKER PARSLEY, HERB MARKER ROSEMARY)	(HERB MARKER THYME)	0.011089	0.012321	0.010553	0.951691	77.240055	0.010417	20.444951
2300	(HERB MARKER THYME, HERB MARKER PARSLEY)	(HERB MARKER ROSEMARY)	0.011089	0.012375	0.010553	0.951691	76.905682	0.010416	20.443842

OUTPUT (using WEKA)



Conclusion –

Implemented Apriori Algorithm using python as well as WEKA software.