**Practical no. 8**

import pandas as pd

from mlxtend.frequent\_patterns import apriori, association\_rules

from mlxtend.preprocessing import TransactionEncoder

import plotly.graph\_objects as go

from plotly.offline import plot

data = [

[1, 21, 'Female', 'L', 'Kurtiset', 'Crape', 1000, 'Printed', 5],

[2, 22, 'Female', 'XL', 'Anarkali', 'Cotton', 2000, 'Plane', 3],

[3, 19, 'Male', 'L', 'Shirt', 'Rayon', 1500, 'Printed', 4],

[4, 10, 'Male', 'S', 'Tshirt', 'Rayon', 500, 'Printed', 4],

[5, 28, 'Male', 'XXL', 'Jeans', 'Denim', 1500, 'Plane', 5],

[6, 18, 'Female', 'S', 'Skirt', 'Rayon', 700, 'Printed', 4],

[7, 30, 'Male', 'L', 'Kurtapajama', 'Cotton', 3000, 'Printed', 5],

[8, 45, 'Female', 'XXL', 'Shirt', 'Rayon', 1500, 'Printed', 3],

[9, 41, 'Female', 'XL', 'Anarkali', 'Satin', 5000, 'Printed', 4],

[10, 23, 'Female', 'XL', 'Jacket', 'Jersey', 8000, 'Plane', 5],

[11, 27, 'Female', 'L', 'Anarkali', 'Cotton', 2000, 'Plane', 4],

[12, 20, 'Female', 'XXL', 'Kurtiset', 'Rayon', 2100, 'Printed', 2],

[13, 32, 'Male', 'L', 'Kurtapajama', 'Denim', 3500, 'Printed', 5],

[14, 24, 'Female', 'XL', 'Jacket', 'Cotton', 2000, 'Printed', 3],

[15, 23, 'Female', 'L', 'Tshirt', 'Satin', 400, 'Plane', 2],

[16, 40, 'Male', 'XL', 'Kurtapajama', 'Rayon', 3000, 'Printed', 5],

[17, 24, 'Female', 'S', 'Jacket', 'Cotton', 2000, 'Printed', 3],

[18, 41, 'Female', 'XL', 'Jeans', 'Denim', 2200, 'Plane', 5],

[19, 29, 'Female', 'XXL', 'Kurtiset', 'Crape', 4000, 'Plane', 3],

[20, 30, 'Female', 'L', 'Jacket', 'Cotton', 2900, 'Plane', 4],

[21, 35, 'Male', 'XL', 'Kurtapajama', 'Satin', 3200, 'Printed', 4],

[22, 25, 'Male', 'L', 'Shirt', 'Denim', 1800, 'Printed', 4],

[23, 50, 'Female', 'XXL', 'Kurtiset', 'Crape', 4500, 'Plane', 3],

[24, 38, 'Male', 'S', 'Tshirt', 'Jersey', 600, 'Printed', 4],

[25, 22, 'Female', 'L', 'Jeans', 'Denim', 2500, 'Plane', 3],

[26, 44, 'Male', 'XL', 'Shirt', 'Cotton', 1800, 'Printed', 5],

[27, 29, 'Female', 'L', 'Kurtiset', 'Rayon', 2100, 'Printed', 5],

[28, 33, 'Male', 'XXL', 'Anarkali', 'Satin', 4000, 'Plane', 2],

[29, 26, 'Female', 'S', 'Kurtapajama', 'Denim', 2600, 'Printed', 5],

[30, 31, 'Female', 'L', 'Shirt', 'Rayon', 1700, 'Printed', 4],

[31, 23, 'Male', 'XL', 'Jeans', 'Rayon', 1900, 'Plane', 3],

[32, 40, 'Female', 'XL', 'Kurtiset', 'Satin', 2200, 'Printed', 4],

[33, 21, 'Male', 'XXL', 'Skirt', 'Crape', 700, 'Plane', 5],

[34, 39, 'Female', 'L', 'Anarkali', 'Rayon', 3500, 'Printed', 5],

[35, 50, 'Male', 'S', 'Jacket', 'Cotton', 2100, 'Printed', 3],

[36, 28, 'Male', 'XXL', 'Kurtiset', 'Rayon', 2700, 'Plane', 5],

[37, 41, 'Female', 'L', 'Jeans', 'Denim', 2300, 'Printed', 3],

[38, 34, 'Female', 'XL', 'Kurtapajama', 'Crape', 4000, 'Plane', 5],

[39, 19, 'Male', 'L', 'Tshirt', 'Satin', 900, 'Printed', 4],

[40, 26, 'Female', 'S', 'Kurtapajama', 'Denim', 2200, 'Plane', 4],

[41, 23, 'Male', 'XL', 'Kurtiset', 'Jersey', 3100, 'Printed', 5],

[42, 29, 'Female', 'XXL', 'Tshirt', 'Rayon', 1800, 'Plane', 3],

[43, 24, 'Male', 'L', 'Shirt', 'Cotton', 1500, 'Printed', 4],

[44, 38, 'Female', 'L', 'Kurtapajama', 'Rayon', 3300, 'Plane', 5],

[45, 30, 'Male', 'XXL', 'Jacket', 'Denim', 5000, 'Printed', 5],

[46, 36, 'Female', 'XL', 'Skirt', 'Satin', 2800, 'Plane', 2],

[47, 22, 'Female', 'S', 'Kurtiset', 'Cotton', 1400, 'Printed', 5],

[48, 25, 'Male', 'L', 'Kurtapajama', 'Satin', 2900, 'Printed', 3],

[49, 42, 'Female', 'L', 'Tshirt', 'Denim', 1500, 'Plane', 4],

[50, 33, 'Male', 'XXL', 'Jeans', 'Crape', 2100, 'Printed', 3]

]

df = pd.DataFrame(data, columns=["ID", "Age", "Gender", "Size", "Product", "Material", "Price", "Type", "Rating"])

transaction\_data = df[['Size', 'Product', 'Material', 'Type']].values.tolist()

te = TransactionEncoder()

te\_ary = te.fit\_transform(transaction\_data)

df\_encoded = pd.DataFrame(te\_ary, columns=te.columns\_)

frequent\_itemsets = apriori(df\_encoded, min\_support=0.1, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.5)

rules['antecedents'] = rules['antecedents'].apply(lambda x: str(x))

rules['consequents'] = rules['consequents'].apply(lambda x: str(x))

fig = go.Figure(data=go.Scatter(

x=rules['support'],

y=rules['confidence'],

mode='markers',

marker=dict(

size=rules['lift'] \* 10, # Bubble size proportional to 'lift'

color=rules['lift'], # Color based on 'lift'

colorscale='Viridis',

showscale=True

),

text=rules['antecedents'] + ' --> ' + rules['consequents'],

hoverinfo='text'

))

fig.update\_layout(

title="Association Rules Visualization",

xaxis\_title="Support",

yaxis\_title="Confidence",

template="plotly\_dark"

)

plot(fig, filename="association\_rules\_visualization.html")

print("Offline plot saved as 'association\_rules\_visualization.html'")

**Output:**



