**使用mlxtend包**

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

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import networkx as nx

# 读取数据

user\_goods = pd.read\_excel('goods\_new.xls', header=None)

# 数据预处理为0-1矩阵

ct = lambda x: pd.Series(1, index=x[pd.notnull(x)])

b = map(ct, user\_goods.values)

data = pd.DataFrame(list(b)).fillna(0).astype(int)

# 将数据转换为布尔类型

data = data.astype(bool)

# Apriori算法

frequent\_itemsets = apriori(data, min\_support=0.2, use\_colnames=True)

# 关联规则分析

rules = association\_rules(frequent\_itemsets, metric="confidence", min\_threshold=0.5, num\_itemsets=len(frequent\_itemsets))

# 输出结果

rules.to\_excel('apriori\_rules.xlsx', index=False)

print("Apriori规则分析结果已保存为 'apriori\_rules.xlsx'")

# 1. 散点图: 支持度 vs 置信度，颜色表示提升度

plt.figure(figsize=(10, 6))

plt.scatter(rules['support'], rules['confidence'], c=rules['lift'], cmap='viridis', s=100)

plt.colorbar(label='Lift')

plt.xlabel('Support')

plt.ylabel('Confidence')

plt.title('Support vs Confidence (Color: Lift)')

plt.grid(True)

plt.show()

# 2. 频繁项集支持度柱状图

top\_items = frequent\_itemsets.nlargest(10, 'support')

plt.figure(figsize=(10, 6))

plt.barh(top\_items['itemsets'].astype(str), top\_items['support'], color='skyblue')

plt.xlabel('Support')

plt.ylabel('Itemsets')

plt.title('Top 10 Frequent Itemsets')

plt.gca().invert\_yaxis()

plt.show()

# 3. 关联规则热力图 (基于提升度)

pivot = rules.pivot(index='antecedents', columns='consequents', values='lift')

pivot.fillna(0, inplace=True)

plt.figure(figsize=(12, 8))

sns.heatmap(pivot, annot=True, fmt=".2f", cmap="YlGnBu", cbar=True)

plt.title('Heatmap of Lift between Antecedents and Consequents')

plt.show()

# 4. 规则网络图 (展示前10个关联规则)

G = nx.DiGraph()

# 添加节点和边

for \_, row in rules.head(10).iterrows():

for antecedent in row['antecedents']:

for consequent in row['consequents']:

G.add\_edge(antecedent, consequent, weight=row['lift'])

# 绘制网络图

plt.figure(figsize=(12, 8))

pos = nx.spring\_layout(G, k=0.5)

nx.draw\_networkx\_nodes(G, pos, node\_size=700, node\_color='lightblue')

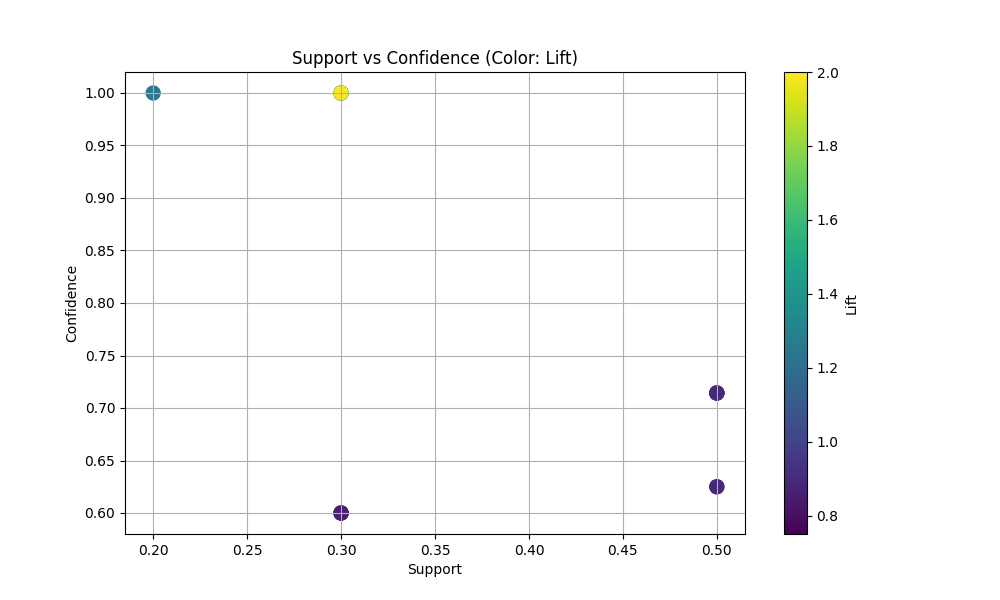
nx.draw\_networkx\_edges(G, pos, arrowstyle='->', arrowsize=20, edge\_color='gray')

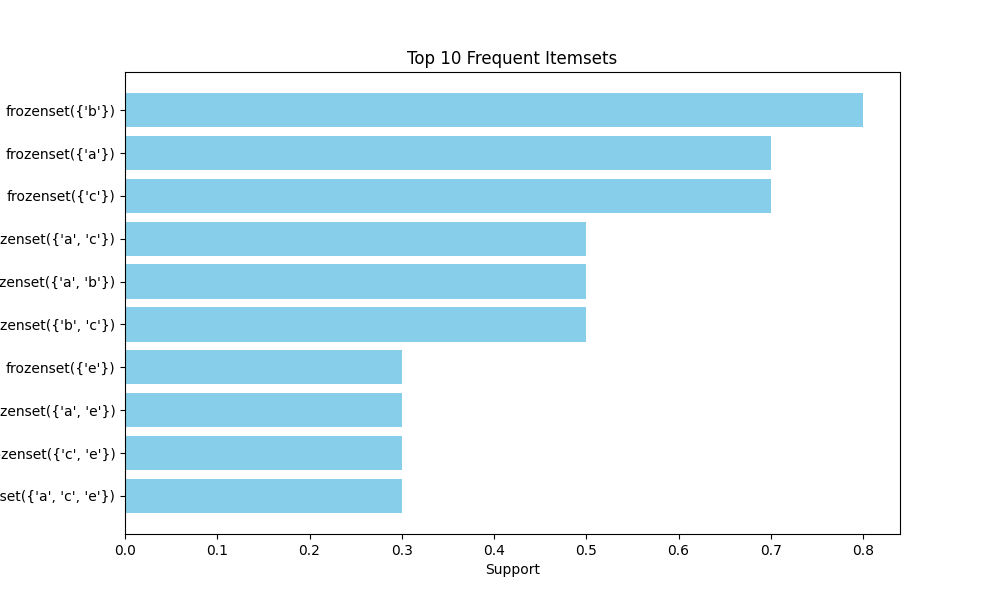
nx.draw\_networkx\_labels(G, pos, font\_size=10, font\_color='black')

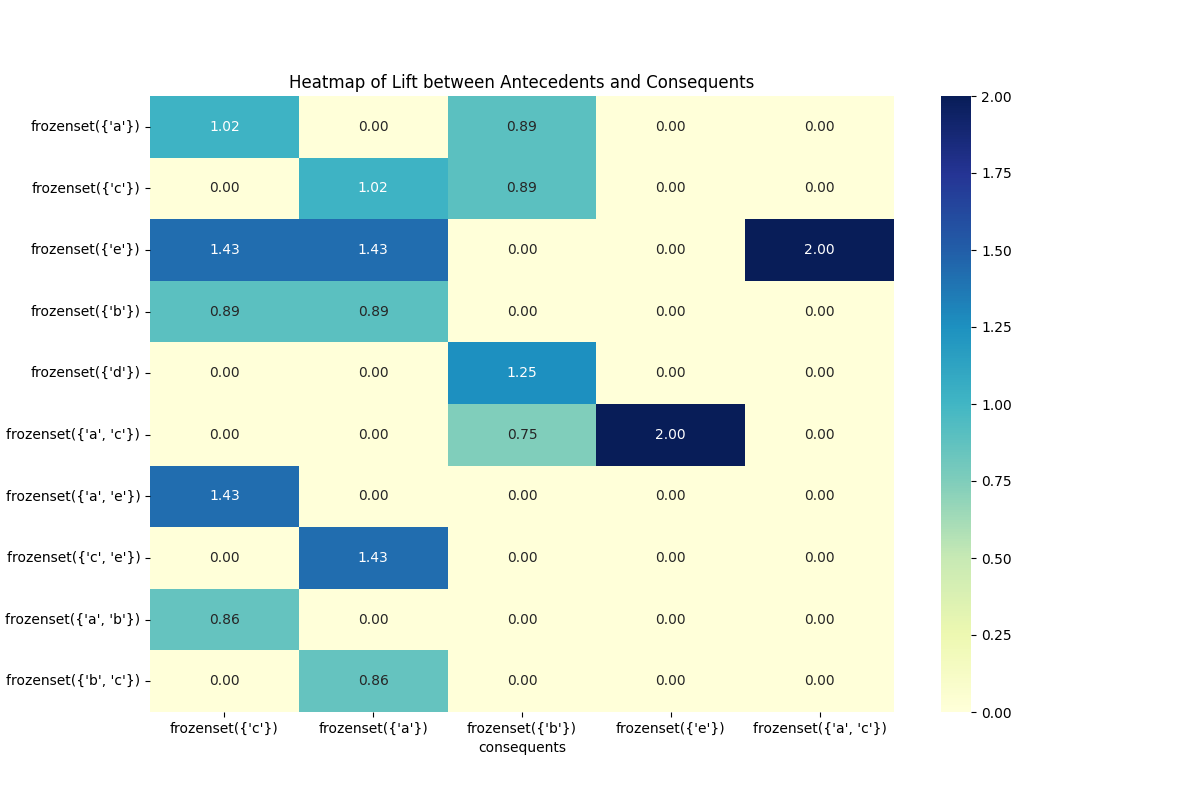
plt.title('Top 10 Association Rules Network Graph')

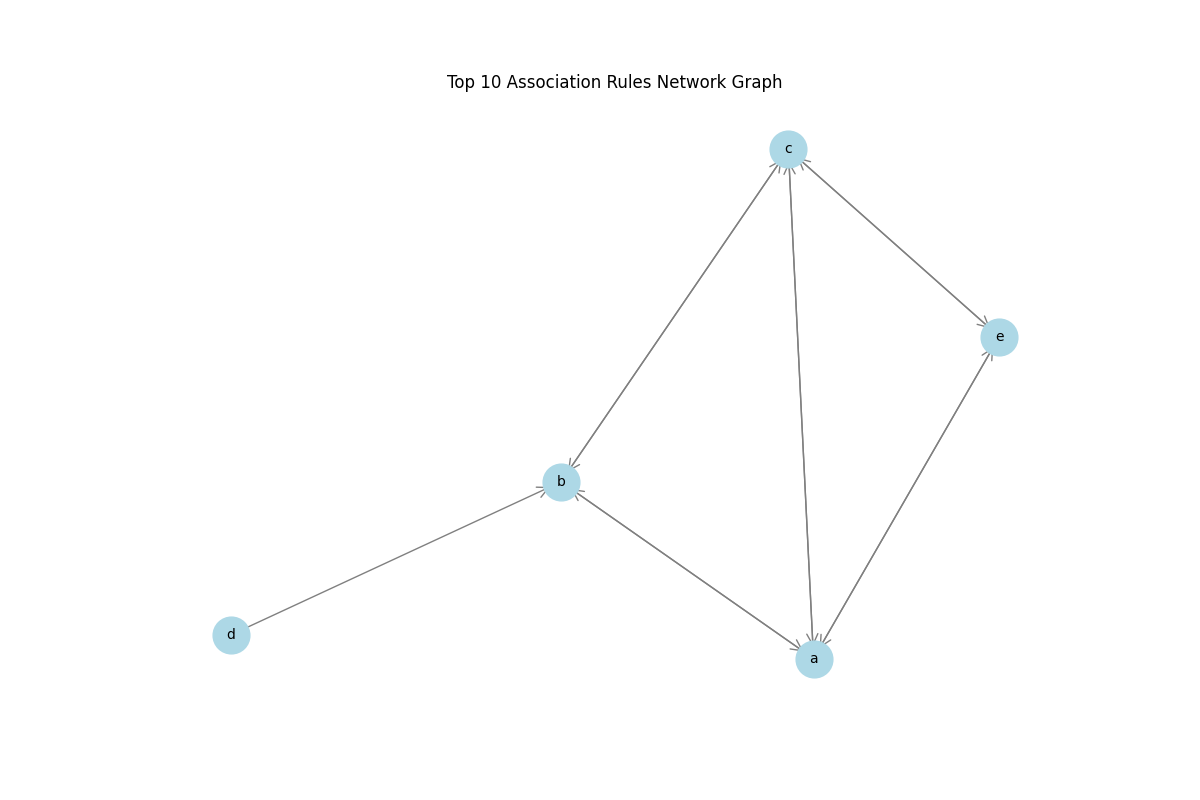
plt.axis('off')

plt.show()









**使用FP-Growth算法**

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

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import networkx as nx

# 读取数据

user\_goods = pd.read\_excel('goods\_new.xls', header=None)

# 数据预处理为0-1矩阵

ct = lambda x: pd.Series(1, index=x[pd.notnull(x)])

b = map(ct, user\_goods.values)

data = pd.DataFrame(list(b)).fillna(0).astype(int)

# 将数据转换为布尔类型

data = data.astype(bool)

# Apriori算法

frequent\_itemsets = apriori(data, min\_support=0.2, use\_colnames=True)

# 关联规则分析

rules = association\_rules(frequent\_itemsets, metric="confidence", min\_threshold=0.5, num\_itemsets=len(frequent\_itemsets))

# 输出结果

rules.to\_excel('apriori\_rules.xlsx', index=False)

print("Apriori规则分析结果已保存为 'apriori\_rules.xlsx'")

# 1. 散点图: 支持度 vs 置信度，颜色表示提升度

plt.figure(figsize=(10, 6))

plt.scatter(rules['support'], rules['confidence'], c=rules['lift'], cmap='viridis', s=100)

plt.colorbar(label='Lift')

plt.xlabel('Support')

plt.ylabel('Confidence')

plt.title('Support vs Confidence (Color: Lift)')

plt.grid(True)

plt.show()

# 2. 频繁项集支持度柱状图

top\_items = frequent\_itemsets.nlargest(10, 'support')

plt.figure(figsize=(10, 6))

plt.barh(top\_items['itemsets'].astype(str), top\_items['support'], color='skyblue')

plt.xlabel('Support')

plt.ylabel('Itemsets')

plt.title('Top 10 Frequent Itemsets')

plt.gca().invert\_yaxis()

plt.show()

# 3. 关联规则热力图 (基于提升度)

pivot = rules.pivot(index='antecedents', columns='consequents', values='lift')

pivot.fillna(0, inplace=True)

plt.figure(figsize=(12, 8))

sns.heatmap(pivot, annot=True, fmt=".2f", cmap="YlGnBu", cbar=True)

plt.title('Heatmap of Lift between Antecedents and Consequents')

plt.show()

# 4. 规则网络图 (展示前10个关联规则)

G = nx.DiGraph()

# 添加节点和边

for \_, row in rules.head(10).iterrows():

for antecedent in row['antecedents']:

for consequent in row['consequents']:

G.add\_edge(antecedent, consequent, weight=row['lift'])

# 绘制网络图

plt.figure(figsize=(12, 8))

pos = nx.spring\_layout(G, k=0.5)

nx.draw\_networkx\_nodes(G, pos, node\_size=700, node\_color='lightblue')

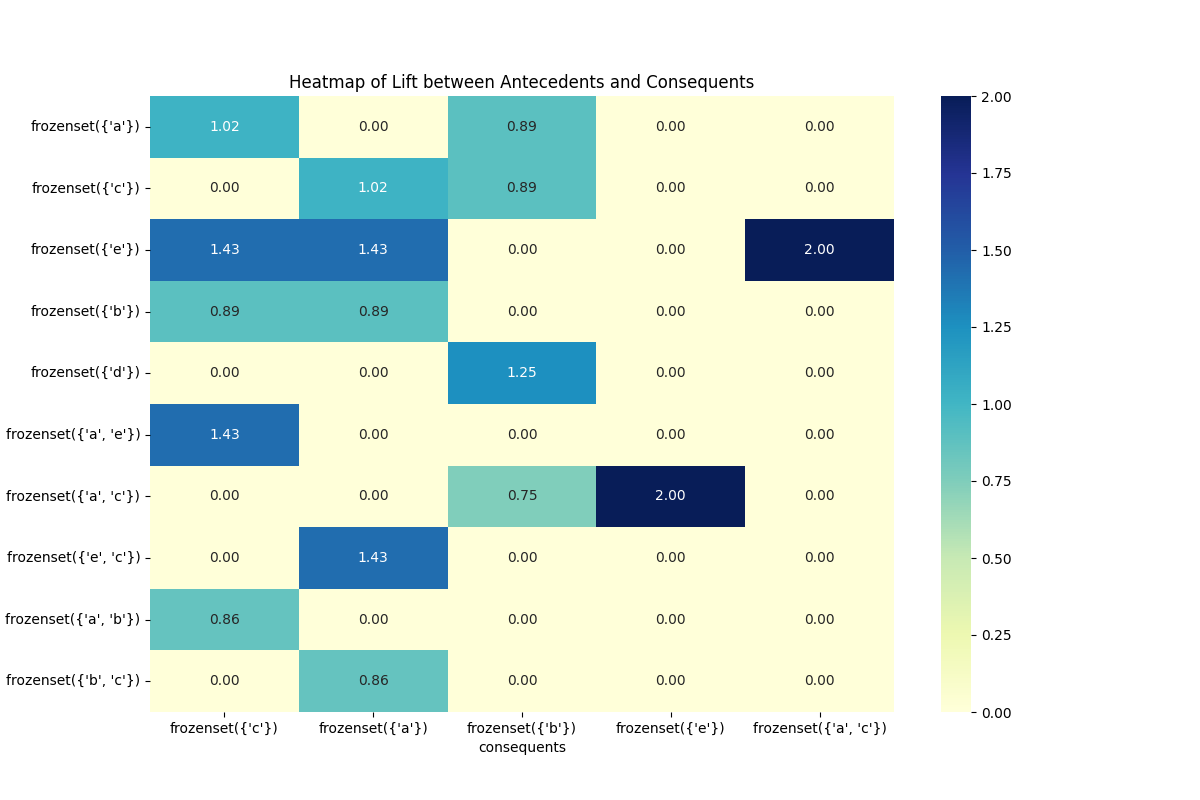
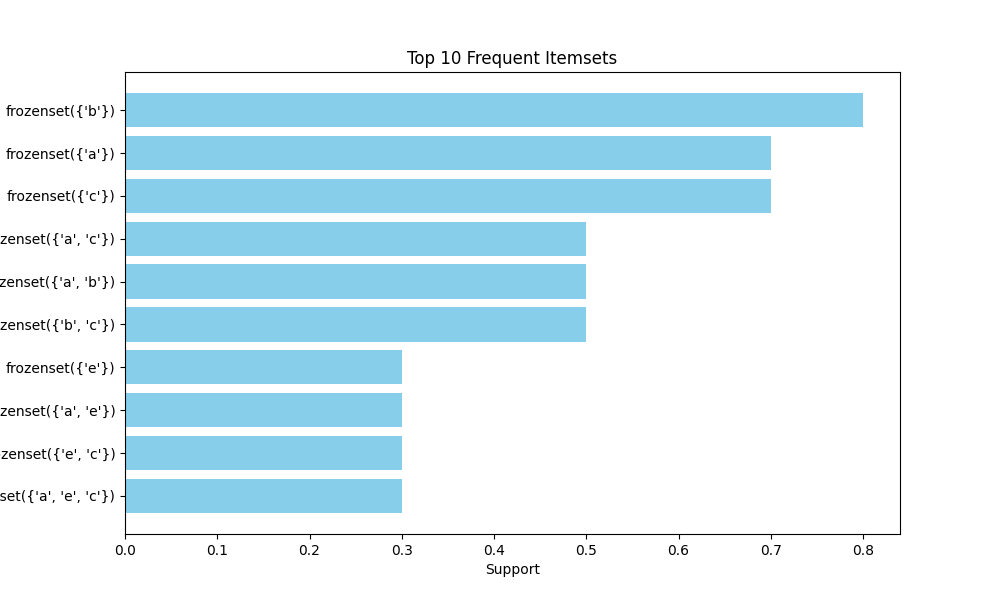
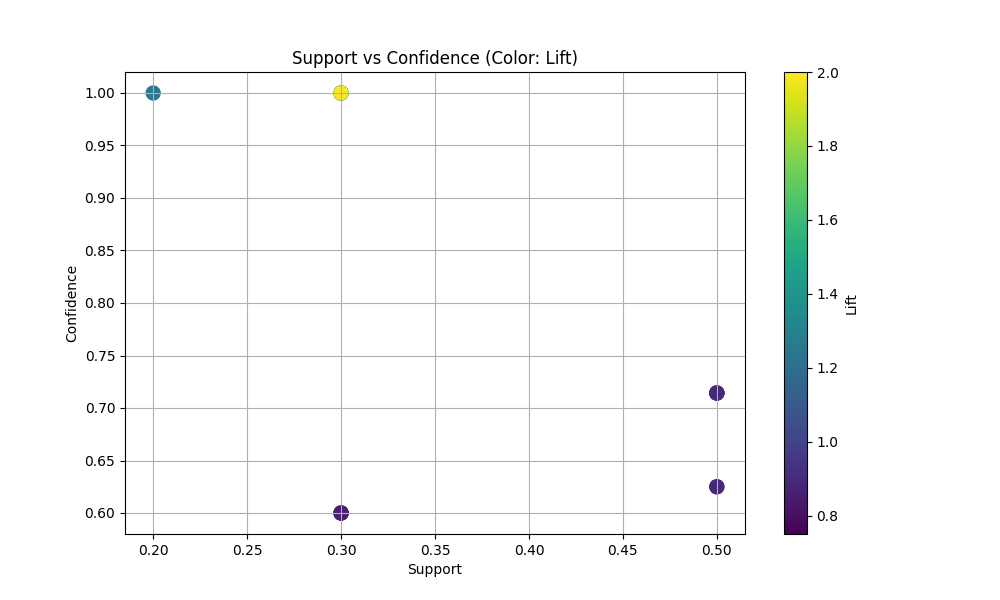
nx.draw\_networkx\_edges(G, pos, arrowstyle='->', arrowsize=20, edge\_color='gray')

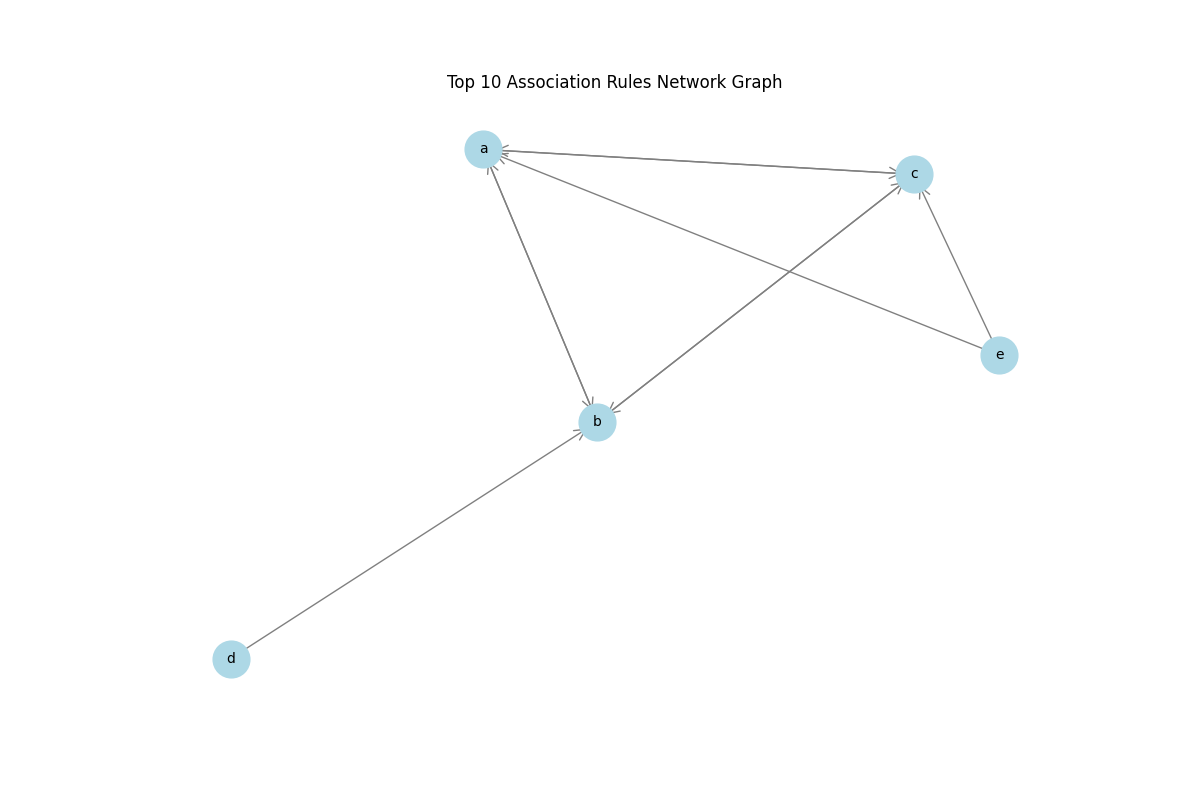
nx.draw\_networkx\_labels(G, pos, font\_size=10, font\_color='black')

plt.title('Top 10 Association Rules Network Graph')

plt.axis('off')

plt.show()

****

****