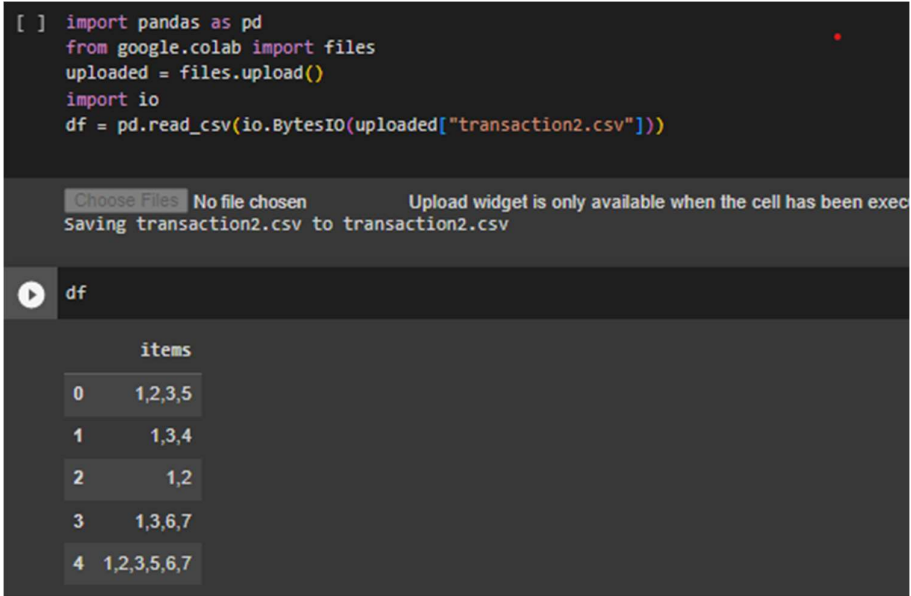


Semester	T.E. Semester V – Computer Engineering
Subject	Data Warehousing and Mining
Subject Professor In-charge	Prof. Kavita Shirsat
Assisting Teachers	Prof. Kavita Shirsat
Laboratory	Lab 312 A

Student Name	Deep Salunkhe
Roll Number	21102A0014
Grade and Subject	
Teacher's Signature	

Experiment Number	07	
Experiment Title	Implementation of Apriori algorithm.	
Resources / Apparatus Required	Hardware: Computer system	Software: Python
Description	<p>The Apriori algorithm is a widely used algorithm in data mining and association rule learning. It is used to discover frequent itemsets in a transaction database and generate association rules based on these frequent itemsets. Here's a theoretical overview of the Apriori algorithm:</p> <ul style="list-style-type: none">• Introduction: The Apriori algorithm is designed to find associations or relationships between items in a transactional database.• Objective: Its primary goal is to identify frequent itemsets, which are sets of items that often appear together in transactions.• Support: The algorithm uses a user-defined threshold called "support" to determine the minimum frequency required for an itemset to be considered frequent.• Apriori Principle: The algorithm is based on the Apriori principle, which states that if an itemset is frequent, then all of its subsets must also be frequent.• Algorithm Steps:<ol style="list-style-type: none">1. Start with individual items as 1-itemsets.2. Count the support of each 1-itemset by scanning the database.3. Prune infrequent 1-itemsets.4. Generate 2-itemsets from the remaining 1-itemsets.	

	<ol style="list-style-type: none"> 5. Count the support of each 2-itemset. 6. Prune infrequent 2-itemsets. <p>Continue this process, incrementally generating larger itemsets and pruning until no more frequent itemsets can be found.</p> <ul style="list-style-type: none"> • Termination: The algorithm terminates when no more frequent itemsets can be generated. • Association Rule Generation: After identifying frequent itemsets, the Apriori algorithm can generate association rules. These rules express relationships between items, e.g., "If A and B are bought, then C is also bought." 												
Program	 <pre>[] import pandas as pd from google.colab import files uploaded = files.upload() import io df = pd.read_csv(io.BytesIO(uploaded["transaction2.csv"]))</pre> <p>Choose Files No file chosen Upload widget is only available when the cell has been executed Saving transaction2.csv to transaction2.csv</p> <p>df</p> <table border="1"> <thead> <tr> <th></th> <th>items</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1,2,3,5</td> </tr> <tr> <td>1</td> <td>1,3,4</td> </tr> <tr> <td>2</td> <td>1,2</td> </tr> <tr> <td>3</td> <td>1,3,6,7</td> </tr> <tr> <td>4</td> <td>1,2,3,5,6,7</td> </tr> </tbody> </table>		items	0	1,2,3,5	1	1,3,4	2	1,2	3	1,3,6,7	4	1,2,3,5,6,7
	items												
0	1,2,3,5												
1	1,3,4												
2	1,2												
3	1,3,6,7												
4	1,2,3,5,6,7												

```

min_support = int(input("Enter minimum no of support count:"))
lst = []
st = set()
for i in df["items"]:
    k = list(map(int,i.split(",")))
    lst.append(k)
    for j in k:
        st.add(j)
c1={}
for i in st:
    c1[i]=sum(i in item for item in lst)
print("\nC1\n1-Itemsets\tsupport count")
for i in c1:
    print(i,"\t\t",c1[i])
c2={}
for i in c1:
    for j in c1:
        if(j>i):
            c2[str(i)+","+str(j)]=None
for i in c2:
    current = list(map(int,i.split(",")))
    c2[i]=sum(current[0] in item and current[1] in item for item in lst)
print("\nC2\n2-Itemsets\tsupport count")
for i in c2:
    print(i,"\t\t",c2[i])
l=[]
for i in c2:
    if(c2[i]<min_support):
        l.append(i)
for i in l:
    c2.pop(i)
print("\nC2\n2-Itemsets\tsupport count")

```

```

print("\nC2\n2-Itemsets\tsupport count")
for i in c2:
    print(i,"\t\t",c2[i])
st= set()
for i in c2:
    for j in list(map(int,i.split(","))):
        st.add(j)
c3={"1,2,3":None,"1,2,5":None,"1,3,5":None,"2,3,5":None}
for i in c3:
    current = list(map(int,i.split(",")))
    c3[i]=sum(current[0] in item and current[1] in item and current[2] in item for item in lst)
l=[]
for i in c3:
    if(c3[i]<min_support):
        l.append(i)
print("\nC3\n3-Itemsets\tsupport count")
for i in c3:
    print(i,"\t\t",c3[i])
for i in l:
    c3.pop(i)
print("\nC3\n3-Itemsets\tsupport count")
for i in c3:
    print(i,"\t\t",c3[i])

```

Output

Enter minimum no of support count:3

C1
1-Itemsets support count
1 5
2 3
3 4
4 1
5 2
6 2
7 2

C2
2-Itemsets support count
1,2 3
1,3 4
1,4 1
1,5 2
1,6 2
1,7 2
2,3 2
2,4 0
2,5 2
2,6 1
2,7 1
3,4 1
3,5 2
3,6 2
3,7 2
4,5 0
4,6 0
4,7 0
5,6 1
5,7 1
6,7 2

L2
2-Itemsets support count
1,2 3
1,3 4

C3|n3-Itemsets support count
1,2,3 2
1,2,5 2
1,3,5 2
2,3,5 2

L3
3-Itemsets support count

Conclusion:

The Apriori algorithm is a fundamental tool in data mining for discovering association rules among items in a dataset. It uses support and confidence measures, as well as the Apriori principle, to efficiently identify frequent itemsets and generate meaningful association rules.