Assignment_7_Association_Mining

February 15, 2021

1 CSB352: Data Mining

Instructor : [Dr. Chandra Prakash]

- For more information visit the class website.

2 LAB Assignment 7: Association Mining

Assigning Date : 15-Feb-2021

Due Date: 21-Feb-2021

Student Name:

Roll No :

3 Assignment Instructions

You must save your as **Assignment_NO_Yourname**

Agenda for the Assignment 7 Different statistical algorithms have been developed to implement association rule mining
1. Apriori Algorithm
2. Frequent Pattern Growth

Your source file will most likely end in **.pynb** if you are using a Jupyter notebook; however, it might also end in **.py** if you are using a Python script.

You have to add your name ; roll no; DATE And Time in the Google Colab Instructions section below and print it.

4 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

[ ]:
```
try:
    from google.colab import drive
    %tensorflow_version 2.x
    COLAB = True
```

```
    print("Assignment #")
    print("Note: using Google CoLab")
except:
    print("Assignment #")
    print("Note: not using Google CoLab")
    COLAB = False
```

[ ]:
```
# Print your name and Roll No.
```

[1]:
```
# Print the curent date
```

[ ]:
```
# Print the curent time
```

Problem Statement :

Association rule mining is a technique to identify underlying relations between different items. Take an example of a Super Market where customers can buy variety of items. Usually, there is a pattern in what the customers buy. For instance, mothers with babies buy baby products such as milk and diapers.

Different statistical algorithms have been developed to implement association rule mining:

1. Apriori Algorithm
2. Frequent Pattern Growth

Support

Support refers to the default popularity of an item and can be calculated by finding number of transactions containing a particular item divided by total number of transactions. Suppose we want to find support for item B. This can be calculated as:

Support(B) = (Transactions containing (B))/(Total Transactions)

Confidence

Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by total number of transactions where A is bought. Mathematically, it can be represented as:

Confidence(A→B) = (Transactions containing both (A and B))/(Transactions containing A)

Lift

Lift(A -> B) refers to the increase in the ratio of sale of B when A is sold. Lift(A –> B) can be calculated by dividing Confidence(A -> B) divided by Support(B). Mathematically it can be represented as:

Lift(A→B) = (Confidence (A→B))/(Support (B))

[ ]:
```
# IMPORT Libraries
```

4.0.1 Task 0: Getting to Know Your Data

Read Dataset [L7_Groceries.csv] from the link from LAB 1

[ ]:
```
# *********** WRITE Your CODE HERE  *************
```

How many unique item set is there in the dataset.

[ ]:
```
# *********** WRITE Your CODE HERE  *************
```

Pre-Processing Part

[13]:
```
## Hint: Apply encoding
```

4.1 TASK 1. Apriori Algorithm

You need to Create your own function for Apriori Algorithm.

eg: **my_Aprirori(Dataset,Confidence,Support)**

Step-1: Determine the support of itemsets in the transactional database.

Step-2: Take all supports in the transaction with higher support value than the minimum or selected support value.

Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-4: Sort the rules as the decreasing order of lift.

[ ]:
```
# *********** WRITE Your FUNCTION FOR Apriori Algorithm HERE  *************
```

[ ]:
```
## 1. Which Patterns Are Interesting? What are the Pattern Evaluation Methods.



# 2. Which parameter indicates the strength of a rule over the random␣
↪occurrence of A and B.
```

4.2 TASK 2. Frequent Pattern Growth Algorithm

You need to Create your own function for FP Grwoth Algorithm.

eg: **my_FPG(Dataset,Confidence,Support)**

Step-1: Determine the Frequent Pattern set

Step-2: Build the Order-Item Set.

Step-3: Tree Construction.

Step-4: Conditional Pattern Base.

Step-5: Condidtional Frequent Pattern Tree

Step-6: Frequent Pattern Generated

Step-7: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-8: Sort the rules as the decreasing order of lift.

[ ]:
```
# ********** WRITE Your FUNCTION FOR Frequent Pattern Growth Algorithm HERE
↪**********
```

[ ]:
```
## Which Patterns Are Interesting?
```

4.3 TASK 3: Compare the results of your funcitons for both algorithm with the inbuild/pre-build packages respectively.

[ ]:
```
# ********** WRITE Your CODE HERE  **********
```

5 Your Learning :

[ ]:
```
# Write  here
```

6 Your Observation about the dataset

[ ]:
```
# Write  here
```