DATA MINING

Course: CS634

Midterm Project

Implementing Apriori Algorithm for

Transactions Dataset

Name: Shaswat Dharaiya
Email: srd22@njit.edu

Table of Content

Definition

Apriori algorithm is an algorithm for frequent item set mining and association rule. It is called Apriori because it uses prior knowledge of frequently used itemset.

Steps

Apriori Algorithm, Find Support:

1) Find the support of each item in all the transactions/itemsets.
2) Eliminate the items with support less than minimum support.
3) Increase the item by 1 and make combinations with all the selected items.
4) Repeat steps 1-3 until no more combinations are possible.

Association Rule, Find confidence:

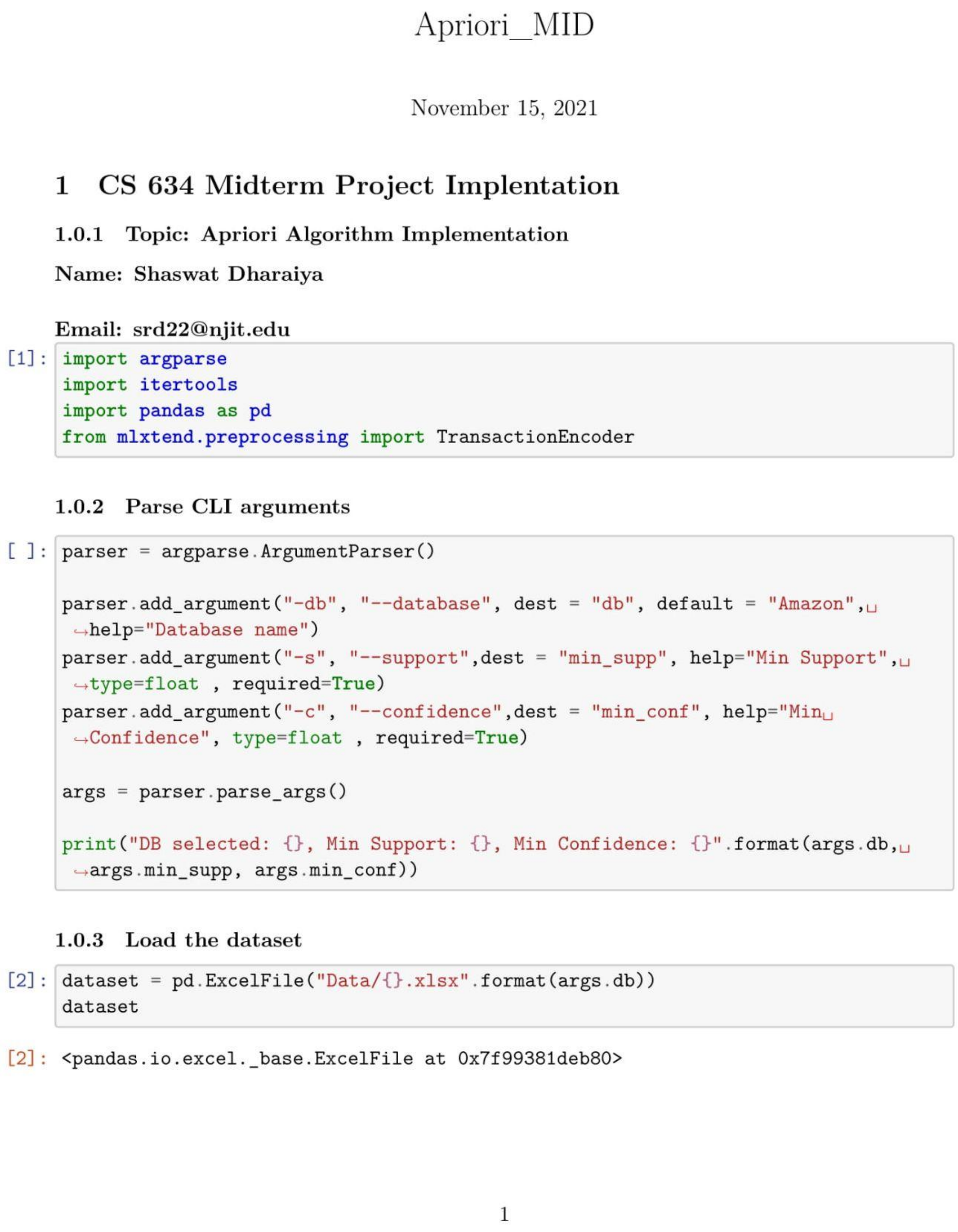
1) From the final combinations of the items, take 1 item and take support of the rest of the items occurring together.
2) Calculate the confidence by dividing the support of all items with the support of the rest of the items occurring together.
3) Repeat steps 1-2 until all confidence for all items is calculated.
4) Repeat steps 1-3 but now increase the number of items by 1 until no more combinations are possible.

Source Code

Programming Language used: Python (v3)

CLI command:

- *Usage:* **python Apriori_MID.py [-h] [-db DB] -s MIN_SUPP -c MIN_CONF**
- *Example:* **python Apriori_MID.py -db Amazon -s 0.5 -c 0.7**



Apriori_MID

November 15, 2021

1 CS 634 Midterm Project Implentation

1.0.1 Topic: Apriori Algorithm Implementation

Name: Shaswat Dharaiya

Email: srd22@njit.edu

[1]:
```
import argparse
import itertools
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
```

1.0.2 Parse CLI arguments

[ ]:
```
parser = argparse.ArgumentParser()

parser.add_argument("-db", "--database", dest = "db", default = "Amazon",␣
 ↪help="Database name")
parser.add_argument("-s", "--support",dest = "min_supp", help="Min Support",␣
 ↪type=float , required=True)
parser.add_argument("-c", "--confidence",dest = "min_conf", help="Min␣
 ↪Confidence", type=float , required=True)

args = parser.parse_args()

print("DB selected: {}, Min Support: {}, Min Confidence: {}".format(args.db,␣
 ↪args.min_supp, args.min_conf))
```

1.0.3 Load the dataset

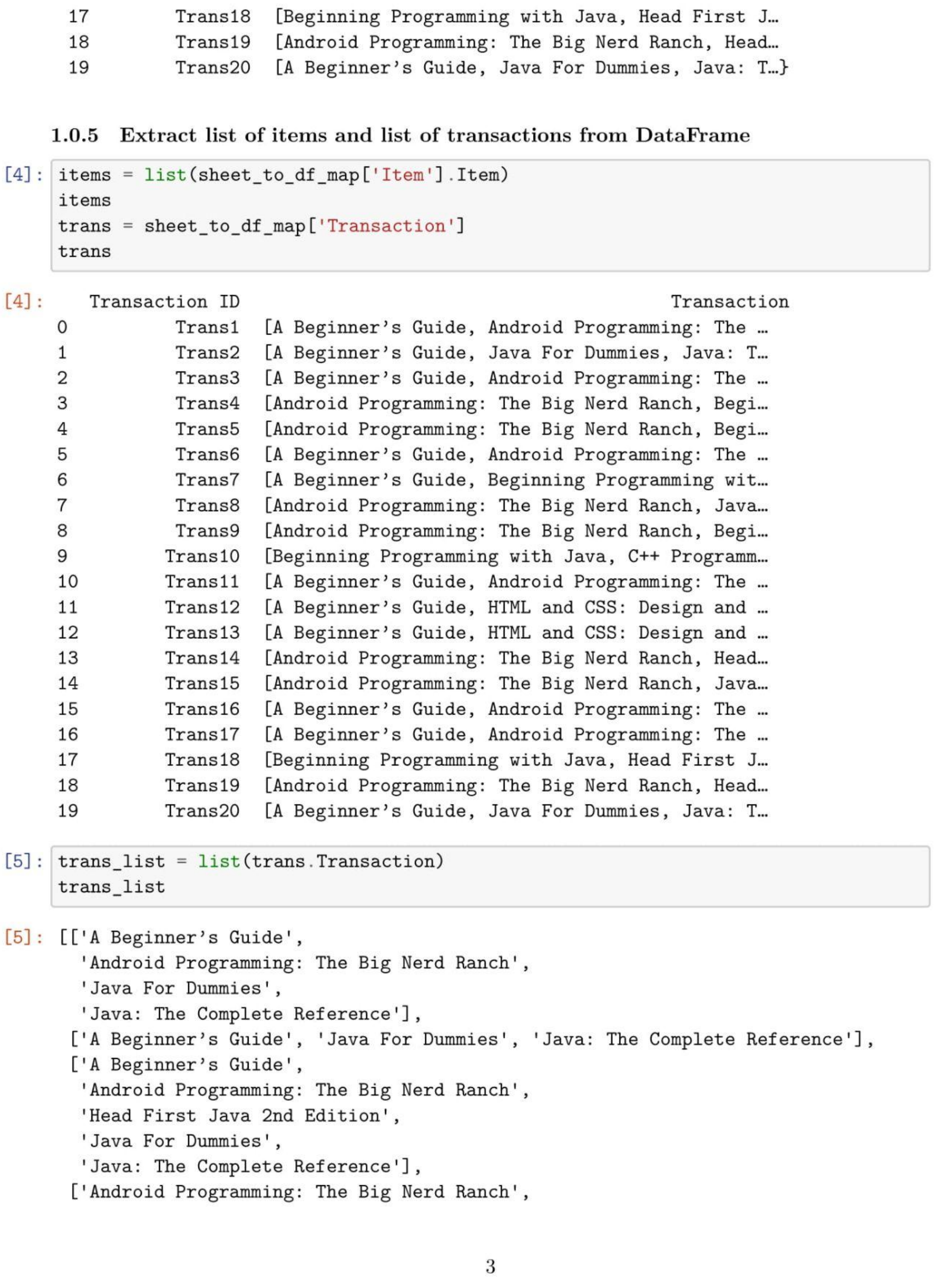
[2]:
```
dataset = pd.ExcelFile("Data/{}.xlsx".format(args.db))
dataset
```

[2]:
```
<pandas.io.excel._base.ExcelFile at 0x7f99381deb80>
```

1.0.4 Load the dataset to dataframe

[3]:
```
sheet_to_df_map = {}
list_of_sheets = dataset.sheet_names
for sheet_name in list_of_sheets:
    indx = "Item #" if "Item" in sheet_name else "Transaction ID"
    cols = dataset.parse(sheet_name).columns
    sheet_to_df_map[sheet_name] = dataset.parse(sheet_name)
    if indx == "Transaction ID":
        sheet_to_df_map[sheet_name]['Transaction'] =␣
↪sheet_to_df_map[sheet_name]['Transaction'].apply(lambda x: sorted([y.strip('␣
↪') for y in x.split(",") if y.strip(' ') != '']))
    else:
        sheet_to_df_map[sheet_name]['Item'] =␣
↪sheet_to_df_map[sheet_name]['Item'].apply(lambda x: x.strip(' '))
sheet_to_df_map
```

[3]:
```
{'Item':    Item #                                     Item
 0       1                       A Beginner’s Guide
 1       2             Java: The Complete Reference
 2       3                         Java For Dummies
 3       4  Android Programming: The Big Nerd Ranch
 4       5              Head First Java 2nd Edition
 5       6          Beginning Programming with Java
 6       7                      Java 8 Pocket Guide
 7       8            C++ Programming in Easy Steps
 8       9             Effective Java (2nd Edition)
 9      10  HTML and CSS: Design and Build Websites,
 'Transaction':    Transaction ID
Transaction
 0          Trans1  [A Beginner’s Guide, Android Programming: The …
 1          Trans2  [A Beginner’s Guide, Java For Dummies, Java: T…
 2          Trans3  [A Beginner’s Guide, Android Programming: The …
 3          Trans4  [Android Programming: The Big Nerd Ranch, Begi…
 4          Trans5  [Android Programming: The Big Nerd Ranch, Begi…
 5          Trans6  [A Beginner’s Guide, Android Programming: The …
 6          Trans7  [A Beginner’s Guide, Beginning Programming wit…
 7          Trans8  [Android Programming: The Big Nerd Ranch, Java…
 8          Trans9  [Android Programming: The Big Nerd Ranch, Begi…
 9         Trans10  [Beginning Programming with Java, C++ Programm…
 10        Trans11  [A Beginner’s Guide, Android Programming: The …
 11        Trans12  [A Beginner’s Guide, HTML and CSS: Design and …
 12        Trans13  [A Beginner’s Guide, HTML and CSS: Design and …
 13        Trans14  [Android Programming: The Big Nerd Ranch, Head…
 14        Trans15  [Android Programming: The Big Nerd Ranch, Java…
 15        Trans16  [A Beginner’s Guide, Android Programming: The …
 16        Trans17  [A Beginner’s Guide, Android Programming: The …
```



```
17        Trans18  [Beginning Programming with Java, Head First J…
18        Trans19  [Android Programming: The Big Nerd Ranch, Head…
19        Trans20  [A Beginner’s Guide, Java For Dummies, Java: T…}
```

1.0.5 Extract list of items and list of transactions from DataFrame

```
[4]: items = list(sheet_to_df_map['Item'].Item)
     items
     trans = sheet_to_df_map['Transaction']
     trans
```

```
[4]:   Transaction ID                                        Transaction
     0          Trans1  [A Beginner’s Guide, Android Programming: The …
     1          Trans2  [A Beginner’s Guide, Java For Dummies, Java: T…
     2          Trans3  [A Beginner’s Guide, Android Programming: The …
     3          Trans4  [Android Programming: The Big Nerd Ranch, Begi…
     4          Trans5  [Android Programming: The Big Nerd Ranch, Begi…
     5          Trans6  [A Beginner’s Guide, Android Programming: The …
     6          Trans7  [A Beginner’s Guide, Beginning Programming wit…
     7          Trans8  [Android Programming: The Big Nerd Ranch, Java…
     8          Trans9  [Android Programming: The Big Nerd Ranch, Begi…
     9         Trans10  [Beginning Programming with Java, C++ Programm…
     10        Trans11  [A Beginner’s Guide, Android Programming: The …
     11        Trans12  [A Beginner’s Guide, HTML and CSS: Design and …
     12        Trans13  [A Beginner’s Guide, HTML and CSS: Design and …
     13        Trans14  [Android Programming: The Big Nerd Ranch, Head…
     14        Trans15  [Android Programming: The Big Nerd Ranch, Java…
     15        Trans16  [A Beginner’s Guide, Android Programming: The …
     16        Trans17  [A Beginner’s Guide, Android Programming: The …
     17        Trans18  [Beginning Programming with Java, Head First J…
     18        Trans19  [Android Programming: The Big Nerd Ranch, Head…
     19        Trans20  [A Beginner’s Guide, Java For Dummies, Java: T…
```

```
[5]: trans_list = list(trans.Transaction)
     trans_list
```

```
[5]: [['A Beginner’s Guide',
       'Android Programming: The Big Nerd Ranch',
       'Java For Dummies',
       'Java: The Complete Reference'],
      ['A Beginner’s Guide', 'Java For Dummies', 'Java: The Complete Reference'],
      ['A Beginner’s Guide',
       'Android Programming: The Big Nerd Ranch',
       'Head First Java 2nd Edition',
       'Java For Dummies',
       'Java: The Complete Reference'],
      ['Android Programming: The Big Nerd Ranch',
 'Beginning Programming with Java',
 'Head First Java 2nd Edition'],
['Android Programming: The Big Nerd Ranch',
 'Beginning Programming with Java',
 'Java 8 Pocket Guide'],
['A Beginner’s Guide',
 'Android Programming: The Big Nerd Ranch',
 'Head First Java 2nd Edition'],
['A Beginner’s Guide',
 'Beginning Programming with Java',
 'Head First Java 2nd Edition'],
['Android Programming: The Big Nerd Ranch',
 'Java For Dummies',
 'Java: The Complete Reference'],
['Android Programming: The Big Nerd Ranch',
 'Beginning Programming with Java',
 'Head First Java 2nd Edition',
 'Java For Dummies'],
['Beginning Programming with Java',
 'C++ Programming in Easy Steps',
 'Java 8 Pocket Guide'],
['A Beginner’s Guide',
 'Android Programming: The Big Nerd Ranch',
 'Java For Dummies',
 'Java: The Complete Reference'],
['A Beginner’s Guide',
 'HTML and CSS: Design and Build Websites',
 'Java For Dummies',
 'Java: The Complete Reference'],
['A Beginner’s Guide',
 'HTML and CSS: Design and Build Websites',
 'Java 8 Pocket Guide',
 'Java For Dummies',
 'Java: The Complete Reference'],
['Android Programming: The Big Nerd Ranch',
 'Head First Java 2nd Edition',
 'Java For Dummies'],
['Android Programming: The Big Nerd Ranch', 'Java For Dummies'],
['A Beginner’s Guide',
 'Android Programming: The Big Nerd Ranch',
 'Java For Dummies',
 'Java: The Complete Reference'],
['A Beginner’s Guide',
 'Android Programming: The Big Nerd Ranch',
 'Java For Dummies',
 'Java: The Complete Reference'],
['Beginning Programming with Java',
 'Head First Java 2nd Edition',
 'Java 8 Pocket Guide'],
['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
['A Beginner’s Guide', 'Java For Dummies', 'Java: The Complete Reference']]
```

1.0.6 Encode the transactions

[6]:
```python
te = TransactionEncoder()
te_data = te.fit(list(trans["Transaction"])).
 ↪transform(list(trans["Transaction"]))
df1 = pd.DataFrame(te_data.astype("int"), columns=te.columns_)
frequent_item = df1.transpose()
```

[7]:
```python
frequent_item
```

[7]:
```
                                          0   1   2   3   4   5   6   7   8  \
A Beginner’s Guide                        1   1   1   0   0   1   1   0   0
Android Programming: The Big Nerd Ranch   1   0   1   1   1   1   0   1   1
Beginning Programming with Java           0   0   0   1   1   0   1   0   1
C++ Programming in Easy Steps             0   0   0   0   0   0   0   0   0
HTML and CSS: Design and Build Websites   0   0   0   0   0   0   0   0   0
Head First Java 2nd Edition               0   0   1   1   0   1   1   0   1
Java 8 Pocket Guide                       0   0   0   0   1   0   0   0   0
Java For Dummies                          1   1   1   0   0   0   0   1   1
Java: The Complete Reference              1   1   1   0   0   0   0   1   0

                                          9  10  11  12  13  14  15  16  17  \
A Beginner’s Guide                        0   1   1   1   0   0   1   1   0
Android Programming: The Big Nerd Ranch   0   1   0   0   1   1   1   1   0
Beginning Programming with Java           1   0   0   0   0   0   0   0   1
C++ Programming in Easy Steps             1   0   0   0   0   0   0   0   0
HTML and CSS: Design and Build Websites   0   0   1   1   0   0   0   0   0
Head First Java 2nd Edition               0   0   0   0   1   0   0   0   1
Java 8 Pocket Guide                       1   0   0   1   0   0   0   0   1
Java For Dummies                          0   1   1   1   1   1   1   1   0
Java: The Complete Reference              0   1   1   1   0   0   1   1   0

                                         18  19
A Beginner’s Guide                        0   1
Android Programming: The Big Nerd Ranch   1   0
Beginning Programming with Java           0   0
C++ Programming in Easy Steps             0   0
HTML and CSS: Design and Build Websites   0   0
Head First Java 2nd Edition               1   0
Java 8 Pocket Guide                       0   0
Java For Dummies                          0   1
Java: The Complete Reference              0   1
```

1.0.7 Get the support of each item

```
[8]: df1.sum()/len(frequent_item.columns)
```

```
[8]: A Beginner’s Guide                           0.55
     Android Programming: The Big Nerd Ranch      0.65
     Beginning Programming with Java              0.30
     C++ Programming in Easy Steps                0.05
     HTML and CSS: Design and Build Websites      0.10
     Head First Java 2nd Edition                  0.40
     Java 8 Pocket Guide                          0.20
     Java For Dummies                             0.65
     Java: The Complete Reference                 0.50
     dtype: float64
```

1.0.8 Get the items with support more than min_supp

```
[9]: min_supp = args.min_supp
     # n*(n-1)/2
     supp = pd.DataFrame(df1.sum()/len(frequent_item.columns), columns=["Support"])
     newlst = sorted(list(supp[supp["Support"]  >= min_supp ].index))
     newlst
```

```
[9]: ['A Beginner’s Guide',
      'Android Programming: The Big Nerd Ranch',
      'Java For Dummies',
      'Java: The Complete Reference']
```

1.0.9 Get combinations of all selected items

```
[10]: def make_combos(lst, key1 = 0):
          combinations = {}
          for L in range(1, len(lst)+1):
              if L != key1:
                  combo = []
                  for subset in itertools.combinations(lst, L):
                      combo.append(list(subset))
                  combinations[L] = combo
          return combinations

      combinations = make_combos(newlst)
      combinations
```

```
[10]: {1: [['A Beginner’s Guide'],
        ['Android Programming: The Big Nerd Ranch'],
        ['Java For Dummies'],
        ['Java: The Complete Reference']],
 2: [['A Beginner’s Guide', 'Android Programming: The Big Nerd Ranch'],
  ['A Beginner’s Guide', 'Java For Dummies'],
  ['A Beginner’s Guide', 'Java: The Complete Reference'],
  ['Android Programming: The Big Nerd Ranch', 'Java For Dummies'],
  ['Android Programming: The Big Nerd Ranch', 'Java: The Complete Reference'],
  ['Java For Dummies', 'Java: The Complete Reference']],
 3: [['A Beginner’s Guide',
   'Android Programming: The Big Nerd Ranch',
   'Java For Dummies'],
  ['A Beginner’s Guide',
   'Android Programming: The Big Nerd Ranch',
   'Java: The Complete Reference'],
  ['A Beginner’s Guide', 'Java For Dummies', 'Java: The Complete Reference'],
  ['Android Programming: The Big Nerd Ranch',
   'Java For Dummies',
   'Java: The Complete Reference']],
 4: [['A Beginner’s Guide',
   'Android Programming: The Big Nerd Ranch',
   'Java For Dummies',
   'Java: The Complete Reference']]}
```

1.0.10 Get combinations that exists in the Transaction list

[11]:
```
new_combo = {}
idx = 0
for key, val in combinations.items():
    count = 0
    new_lst = []
    for lst in val:
        idx += 1
        bools = [ 1 if (set(lst).issubset(set(elem))) else 0 for elem in␣
 ↪trans_list ]
        if not 1 in bools:
            val.remove(lst)
        else:
            count = bools.count(1)
            new_lst.append([lst,count/len(frequent_item.columns)])
        new_combo[idx] = [sorted(lst), count/len(frequent_item.columns)]

final = pd.DataFrame(new_combo.values(),columns=["Items","Support"])
new_combo
```

[11]:
```
{1: [['A Beginner’s Guide'], 0.55],
 2: [['Android Programming: The Big Nerd Ranch'], 0.65],
 3: [['Java For Dummies'], 0.65],
 4: [['Java: The Complete Reference'], 0.5],
 5: [['A Beginner’s Guide', 'Android Programming: The Big Nerd Ranch'], 0.3],
6: [['A Beginner’s Guide', 'Java For Dummies'], 0.45],
7: [['A Beginner’s Guide', 'Java: The Complete Reference'], 0.45],
8: [['Android Programming: The Big Nerd Ranch', 'Java For Dummies'], 0.45],
9: [['Android Programming: The Big Nerd Ranch',
  'Java: The Complete Reference'],
 0.3],
10: [['Java For Dummies', 'Java: The Complete Reference'], 0.5],
11: [['A Beginner’s Guide',
  'Android Programming: The Big Nerd Ranch',
  'Java For Dummies'],
 0.25],
12: [['A Beginner’s Guide',
  'Android Programming: The Big Nerd Ranch',
  'Java: The Complete Reference'],
 0.25],
13: [['A Beginner’s Guide',
  'Java For Dummies',
  'Java: The Complete Reference'],
 0.45],
14: [['Android Programming: The Big Nerd Ranch',
  'Java For Dummies',
  'Java: The Complete Reference'],
 0.3],
15: [['A Beginner’s Guide',
  'Android Programming: The Big Nerd Ranch',
  'Java For Dummies',
  'Java: The Complete Reference'],
 0.25]}
```

1.0.11 Get item sets that have support more than min_supp

[12]:

```
new_final = final[final["Support"] >= min_supp].reset_index(drop=True)
new_final
```

[12]:

```
                                              Items  Support
0                              [A Beginner’s Guide]     0.55
1         [Android Programming: The Big Nerd Ranch]     0.65
2                                [Java For Dummies]     0.65
3                    [Java: The Complete Reference]     0.50
4  [Java For Dummies, Java: The Complete Reference]     0.50
```

[13]:

```
highst_len = 0
if new_final.shape[0] != 0:
    highst_len = len(new_final.Items.iloc[-1])
final_comb = new_final[new_final['Items'].str.len() == highst_len].
 ↪reset_index(drop=True)
final_comb
```

```
[13]:                                          Items  Support
      0  [Java For Dummies, Java: The Complete Reference]      0.5
```

1.0.12 Get the final associations and their corresponding confidence.

```
[15]: cols = ["Item1","Item2","Support1","Support2","Confidence"]
      conf_df = pd.DataFrame(columns=cols)
      for item_lst in final_comb.Items:
          supp_item = new_final[new_final['Items'].apply(lambda x: x == item_lst)].
       ↪values[0][1]
          assc_combo = make_combos(item_lst,highst_len)
          for key, val in assc_combo.items():
              for item in val:
                  item2 = sorted(list(set(item_lst) - set(item)))
                  supp_item2 = new_final[new_final['Items'].apply(lambda x: x ==␣
       ↪item2)].values[0][1]
                  confidence = supp_item / supp_item2
                  if confidence >= args.min_conf:
                      print("{x} -> {y}".format(x=item, y= item2))
                      print("Confidence = Supp({x}) / Supp{y}".format(x=item_lst, y=␣
       ↪item2))
                      print("           = {x} / {y}".format(x=supp_item,␣
       ↪y=supp_item2))
                      print("           = {x:.2f}\n".format(x=confidence))
                  dict_lst = [item,item2,supp_item,supp_item2,confidence]
                  res = {cols[i]: dict_lst[i] for i in range(len(cols))}
                  conf_df = conf_df.append(res, ignore_index=True)
```

```
['Java For Dummies'] -> ['Java: The Complete Reference']
Confidence = Supp(['Java For Dummies', 'Java: The Complete Reference']) /
Supp['Java: The Complete Reference']
           = 0.5 / 0.5
           = 1.00

['Java: The Complete Reference'] -> ['Java For Dummies']
Confidence = Supp(['Java For Dummies', 'Java: The Complete Reference']) /
Supp['Java For Dummies']
           = 0.5 / 0.65
           = 0.77
```

1.0.13 Get support confidence matrix

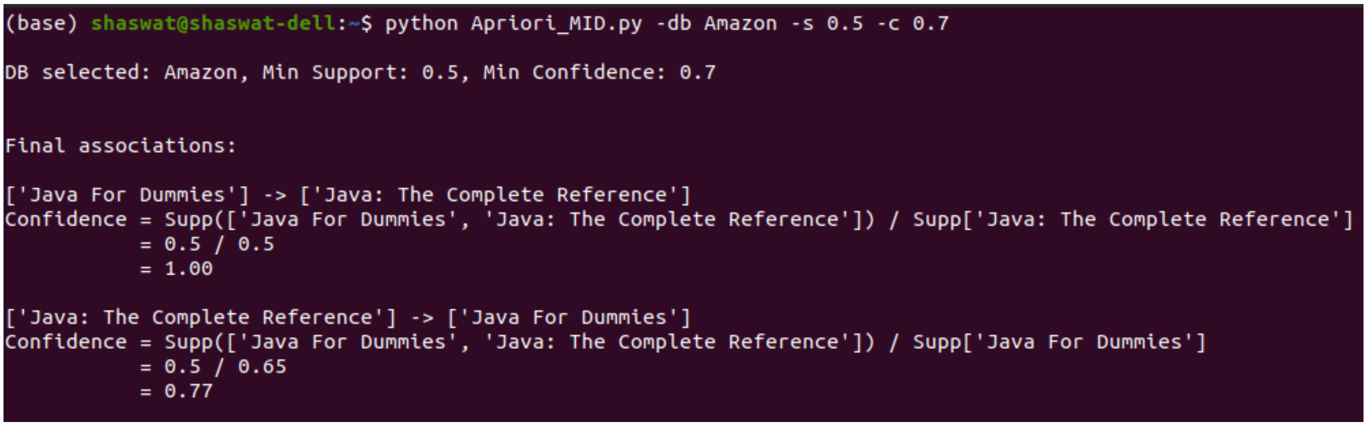
```
[16]: conf_matrix = conf_df[conf_df["Confidence"] >= args.min_conf].
       ↪reset_index(drop=True)
      conf_matrix
```

```
[16]:                          Item1                           Item2  Support1  \
      0             [Java For Dummies]  [Java: The Complete Reference]       0.5
      1  [Java: The Complete Reference]              [Java For Dummies]       0.5

         Support2  Confidence
      0      0.50    1.000000
      1      0.65    0.769231
```

```
[ ]:
```

Output



```
(base) shaswat@shaswat-dell:~$ python Apriori_MID.py -db Amazon -s 0.5 -c 0.7

DB selected: Amazon, Min Support: 0.5, Min Confidence: 0.7


Final associations:

['Java For Dummies'] -> ['Java: The Complete Reference']
Confidence = Supp(['Java For Dummies', 'Java: The Complete Reference']) / Supp['Java: The Complete Reference']
           = 0.5 / 0.5
           = 1.00

['Java: The Complete Reference'] -> ['Java For Dummies']
Confidence = Supp(['Java For Dummies', 'Java: The Complete Reference']) / Supp['Java For Dummies']
           = 0.5 / 0.65
           = 0.77
```