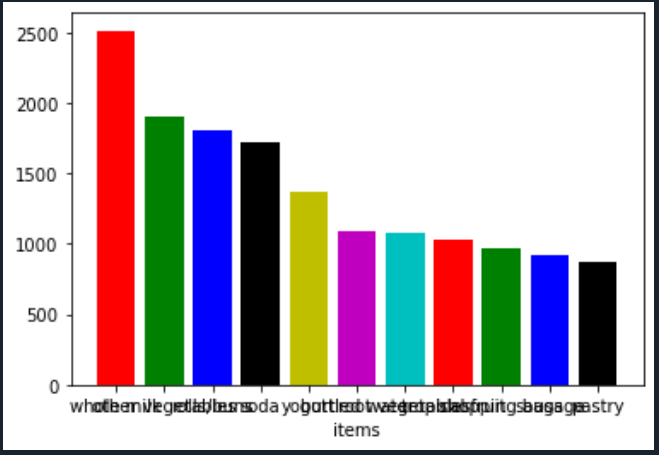
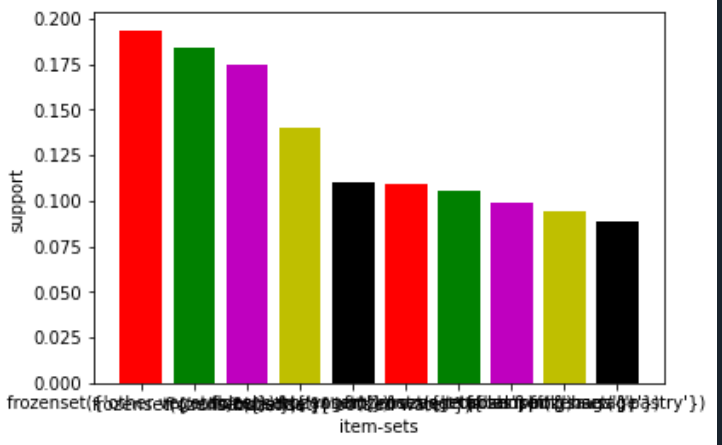
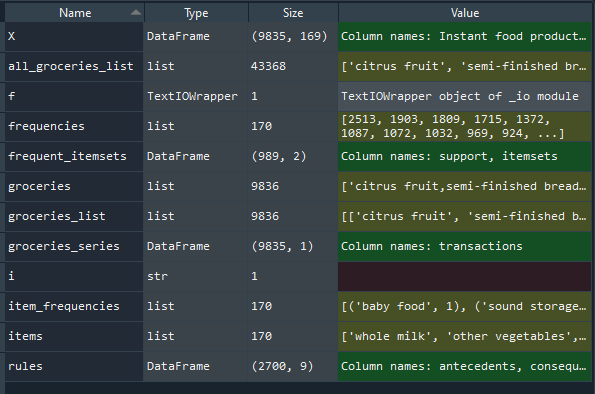
ASSOCIATION RULES

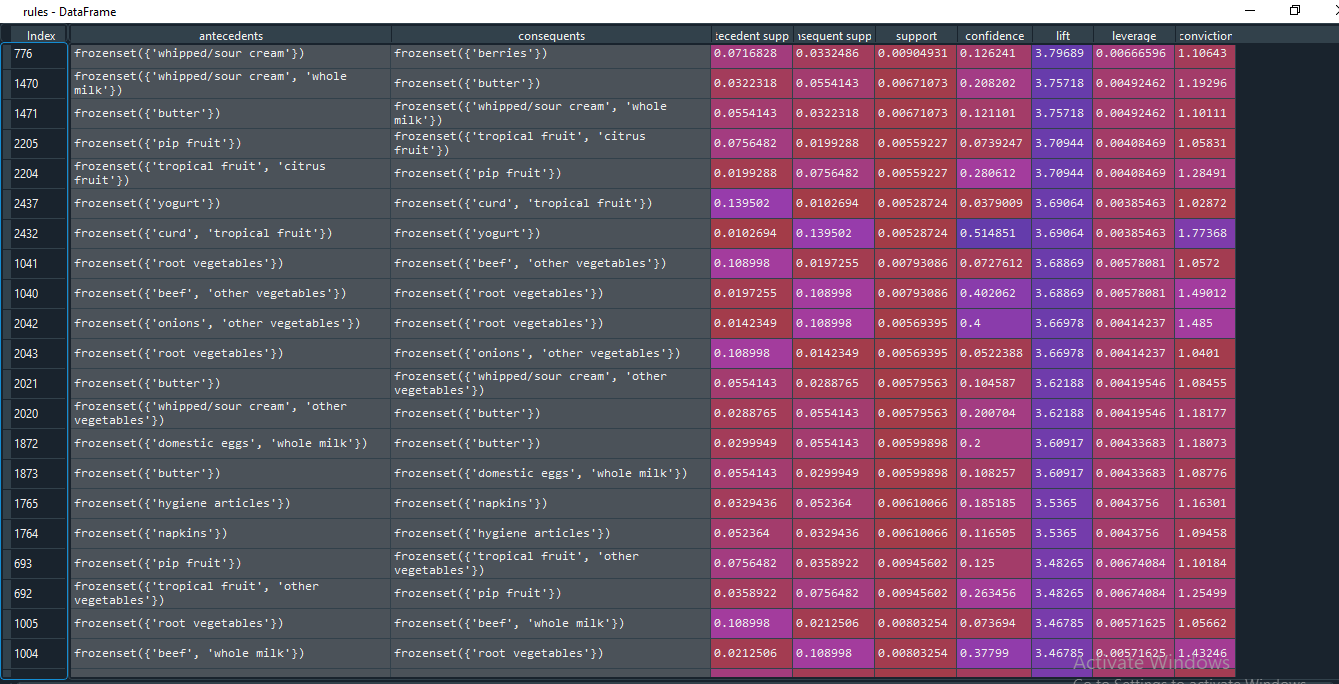
1)for the given data set “groceries.csv”







Rules:



Full code:

# implementing Apriori algorithm from mlxtend

# conda install -c conda-forge mlxtend

import pandas as pd

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

groceries = []

# As the file is in transaction data we will be reading data directly

with open("groceries.csv") as f:

groceries = f.read()

# splitting the data into separate transactions using separator as "\n"

groceries = groceries.split("\n")

groceries\_list = []

for i in groceries:

groceries\_list.append(i.split(","))

all\_groceries\_list = []

#for i in groceries\_list:

# all\_groceries\_list = all\_groceries\_list+i

all\_groceries\_list = [i for item in groceries\_list for i in item]

from collections import Counter

item\_frequencies = Counter(all\_groceries\_list)

# after sorting

#item\_frequencies = sorted(item\_frequencies.items(),key = lambda x:x[1])

item\_frequencies = sorted(item\_frequencies.items(),key = lambda x:x[1])

# Storing frequencies and items in separate variables

frequencies = list(reversed([i[1] for i in item\_frequencies]))

items = list(reversed([i[0] for i in item\_frequencies]))

# barplot of top 10

import matplotlib.pyplot as plt

plt.bar(height = frequencies[0:11],x = list(range(0,11)),color='rgbkymc');plt.xticks(list(range(0,11),),items[0:11]);plt.xlabel("items")

plt.ylabel("Count")

# Creating Data Frame for the transactions data

import pandas as pd

# Purpose of converting all list into Series object Coz to treat each list element as entire element not to separate

groceries\_series = pd.DataFrame(pd.Series(groceries\_list))

groceries\_series = groceries\_series.iloc[:9835,:] # removing the last empty transaction

groceries\_series.columns = ["transactions"]

# creating a dummy columns for the each item in each transactions ... Using column names as item name

X = groceries\_series['transactions'].str.join(sep='\*').str.get\_dummies(sep='\*')

frequent\_itemsets = apriori(X, min\_support=0.005, max\_len=3,use\_colnames = True)

frequent\_itemsets.shape

# Most Frequent item sets based on support

frequent\_itemsets.sort\_values('support',ascending = False,inplace=True)

plt.bar(x = list(range(1,11)),height = frequent\_itemsets.support[1:11],color='rgmyk');plt.xticks(list(range(1,11)),frequent\_itemsets.itemsets[1:11])

plt.xlabel('item-sets');plt.ylabel('support')

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1)

rules.shape

rules.head(1)

rules.sort\_values('lift',ascending = False,inplace=True)

########################## To eliminate Redudancy in Rules ####################################

def to\_list(i):

return (sorted(list(i)))

ma\_X = rules.antecedents.apply(to\_list)+rules.consequents.apply(to\_list)

ma\_X = ma\_X.apply(sorted)

rules\_sets = list(ma\_X)

unique\_rules\_sets = [list(m) for m in set(tuple(i) for i in rules\_sets)]

index\_rules = []

for i in unique\_rules\_sets:

index\_rules.append(rules\_sets.index(i))

# getting rules without any redudancy

rules\_no\_redudancy = rules.iloc[index\_rules,:]

# Sorting them with respect to list and getting top 10 rules

rules\_no\_redudancy.sort\_values('lift',ascending=False).head(10)

2) For the given dataset “book.csv”

FuLL CODE:

# -\*- coding: utf-8 -\*-

"""

Created on Wed Oct 21 18:13:03 2020

@author: sunil

"""

import pandas as pd

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

book = []

# As the file is in transaction data we will be reading data directly

with open("book.csv") as f:

book\_it = f.read()

book\_it= book\_it.split("\n")

book\_list=[]

for i in book\_it:

book\_list.append(i.split(","))

all\_book\_list=[]

all\_book\_list=[i for item in book\_list for i in item]

from collections import Counter

item\_frequencies=Counter(all\_book\_list)

item\_frequencies = sorted(item\_frequencies.items(),key = lambda x:x[1])

frequencies = list(reversed([i[1] for i in item\_frequencies]))

items = list(reversed([i[0] for i in item\_frequencies]))

import matplotlib.pyplot as plt

plt.bar(height = frequencies[1:],x = list(range(0,13)),color='rgbkymc');plt.xticks(list(range(0,13),),items[1:]);plt.xlabel("items")

plt.ylabel("Count")

import pandas as pd

book\_series = pd.DataFrame(pd.Series(book\_list))

book\_series.columns = ["transactions"]

X = book\_series['transactions'].str.join(sep='\*').str.get\_dummies(sep='\*')

frequent\_itemsets = apriori(X, min\_support=0.005, max\_len=3,use\_colnames = True)

frequent\_itemsets.shape

# Most Frequent item sets based on support

frequent\_itemsets.sort\_values('support',ascending = False,inplace=True)

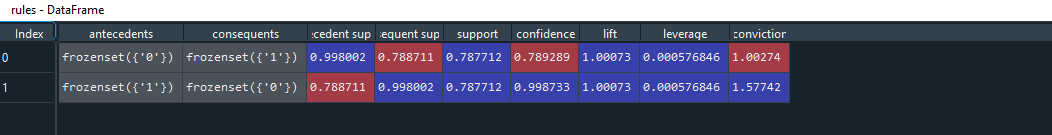
plt.bar(x = list(range(0,3)),height = frequent\_itemsets.support[0:3],color='rgmyk');plt.xticks(list(range(0,3)),frequent\_itemsets.itemsets[0:3])

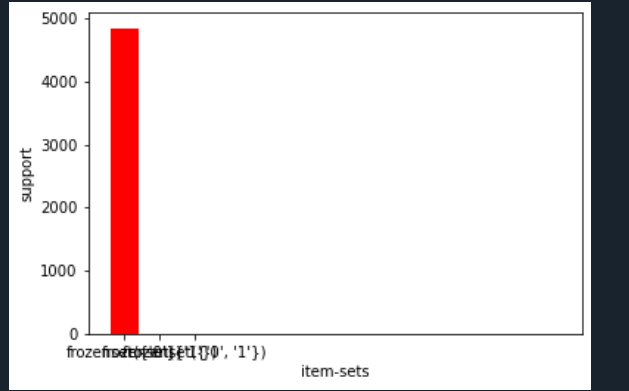
plt.xlabel('item-sets');plt.ylabel('support')

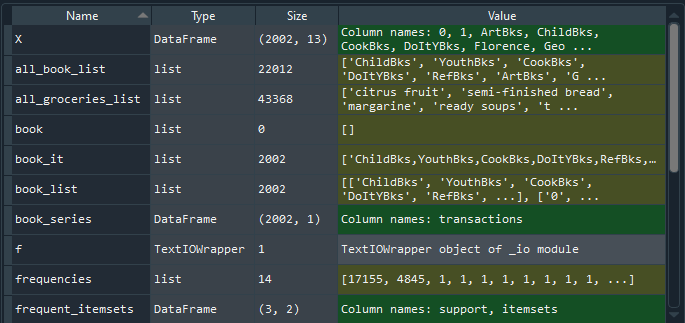
rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1)

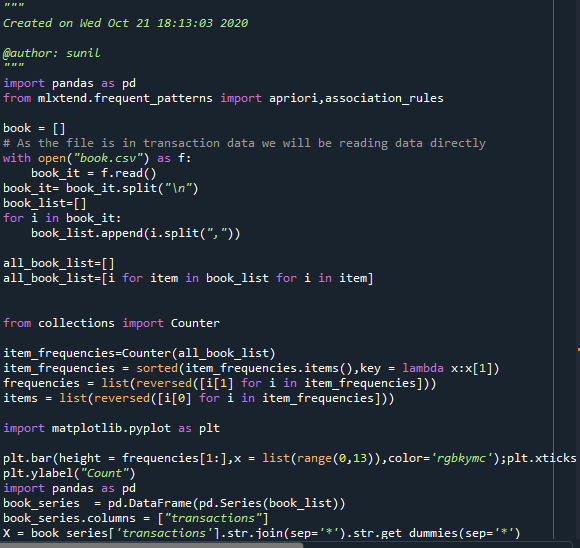
rules.shape

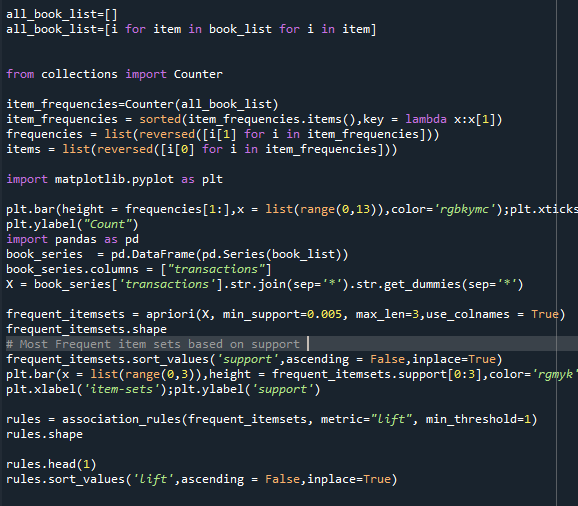
rules.head(1)

rules.sort\_values('lift',ascending = False,inplace=True) 









3) For the given dataset ”my\_movies.csv”

FULL CODE:

# -\*- coding: utf-8 -\*-

"""

Created on Thu Oct 22 09:58:56 2020

@author: sunil

"""

import pandas as pd

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

movies=[]

with open("my\_movies.csv") as f:

movies\_it=f.read()

movies\_it=movies\_it.split("\n")

movies\_list=[]

for i in movies\_it:

movies\_list.append(i.split(","))

all\_movies\_list=[]

all\_movies\_list=[i for item in movies\_list for i in item]

from collections import Counter

item\_frequencies=Counter(all\_movies\_list)

item\_frequencies=sorted(item\_frequencies.items(),key=lambda x:x[1])

frequencies = list(reversed([i[1] for i in item\_frequencies]))

items = list(reversed([i[0] for i in item\_frequencies]))

import matplotlib.pyplot as plt

plt.bar(height = frequencies[1:],x = list(range(0,18)),color='rgbkymc');plt.xticks(list(range(0,18),),items[1:]);plt.xlabel("items")

plt.ylabel("Count")

import pandas as pd

movies\_series = pd.DataFrame(pd.Series(movies\_list))

movies\_series.columns = ["transactions"]

X = movies\_series['transactions'].str.join(sep='\*').str.get\_dummies(sep='\*')

frequent\_itemsets = apriori(X, min\_support=0.005, max\_len=3,use\_colnames = True)

frequent\_itemsets.shape

# Most Frequent item sets based on support

frequent\_itemsets.sort\_values('support',ascending = False,inplace=True)

plt.bar(x = list(range(0,3)),height = frequent\_itemsets.support[0:3],color='rgmyk');plt.xticks(list(range(0,3)),frequent\_itemsets.itemsets[0:3])

plt.xlabel('item-sets');plt.ylabel('support')

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1)

rules.shape

rules.head(1)

rules.sort\_values('lift',ascending = False,inplace=True) 