MarketBasket

April 7, 2019

```
In [1]: import mlxtend
        import numpy as np
        import pandas as pd
        import re

        from mpl_toolkits.mplot3d import Axes3D
        import matplotlib.pyplot as plt
        import pandas as pd
        import seaborn as sns

        from mlxtend.preprocessing import TransactionEncoder
        from mlxtend.frequent_patterns import apriori
        from mlxtend.frequent_patterns import association_rules

        from scipy.stats import chi2_contingency
```

Libary example

```
dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)

frequent_itemsets
```

0.0.1 Importando o datase

full dataset outputed by the data understanding

```
In [2]: ratingsDF = pd.read_pickle("..\\ProcessedData\\df.pkl")
```

0.0.2 Creating users baskets

A biblipteca precisa de uma lista de listas, em que cada linhas estão listados os filmes que um dado usuário viu. Por isso vamos ter que transormar o dataset fazendo uma operação de pivotemanto

```
In [3]: df_pivoted = pd.pivot_table(ratingsDF, columns=['movie_title'], values=['id'], index=[
```

```
In [4]: df_pivoted.fillna(False, inplace=True)
```

```
In [5]: df_pivoted.head(10).head()
```

```
Out[5]:                                    id                                    \
        movie_title 'Til There Was You (1997) 1-900 (1994) 101 Dalmatians (1996)
        uid
        1                                False        False                  True
        2                                False        False                 False
        3                                False        False                 False
        4                                False        False                 False
        5                                False        False                  True

                                                                                \
        movie_title 12 Angry Men (1957) 187 (1997) 2 Days in the Valley (1996)
        uid
        1                           True      False                       False
        2                          False      False                       False
        3                          False       True                       False
        4                          False      False                       False
        5                          False      False                       False

                                                                                  \
        movie_title 20,000 Leagues Under the Sea (1954) 2001: A Space Odyssey (1968)
        uid
        1                                        True                        True
        2                                       False                       False
        3                                       False                       False
        4                                       False                       False
        5                                       False                        True

                                                                                  \
        movie_title 3 Ninjas: High Noon At Mega Mountain (1998) 39 Steps, The (1935)
        uid
        1                                              False               False
        2                                               True               False
        3                                              False               False
        4                                              False               False
        5                                              False               False

                                   ...                                    \
        movie_title                ...                 Yankee Zulu (1994)
```

```
uid                          ...
1                            ...                                False
2                            ...                                False
3                            ...                                False
4                            ...                                False
5                            ...                                False

                                                                    \
movie_title Year of the Horse (1997) You So Crazy (1994)
uid
1                                False              False
2                                False              False
3                                False              False
4                                False              False
5                                False              False

                                                                          \
movie_title Young Frankenstein (1974) Young Guns (1988) Young Guns II (1990)
uid
1                                 True              True                False
2                                False             False                False
3                                False             False                False
4                                False             False                False
5                                 True             False                False

                                                                        \
movie_title Young Poisoner's Handbook, The (1995) Zeus and Roxanne (1997)
uid
1                                              False                   False
2                                              False                   False
3                                              False                   False
4                                              False                   False
5                                              False                   False


movie_title unknown Á köldum klaka (Cold Fever) (1994)
uid
1                True                                False
2               False                                False
3               False                                False
4               False                                False
5                True                                False

[5 rows x 1664 columns]
```

In [6]:
```
df_pivoted["id"].head()
```

Out[6]:
```
movie_title  'Til There Was You (1997)  1-900 (1994)  101 Dalmatians (1996)  \
uid
```

```
1                                False         False                    True
2                                False         False                   False
3                                False         False                   False
4                                False         False                   False
5                                False         False                    True

movie_title  12 Angry Men (1957)  187 (1997)  2 Days in the Valley (1996)  \
uid
1                           True       False                         False
2                          False       False                         False
3                          False        True                         False
4                          False       False                         False
5                          False       False                         False

movie_title  20,000 Leagues Under the Sea (1954)  \
uid
1                                          True
2                                         False
3                                         False
4                                         False
5                                         False

movie_title  2001: A Space Odyssey (1968)  \
uid
1                                    True
2                                   False
3                                   False
4                                   False
5                                    True

movie_title  3 Ninjas: High Noon At Mega Mountain (1998)  \
uid
1                                              False
2                                               True
3                                              False
4                                              False
5                                              False

movie_title  39 Steps, The (1935)                ...                   \
uid                                              ...
1                           False                ...
2                           False                ...
3                           False                ...
4                           False                ...
5                           False                ...

movie_title  Yankee Zulu (1994)  Year of the Horse (1997)  \
uid
```

```
1                          False                     False
2                          False                     False
3                          False                     False
4                          False                     False
5                          False                     False

movie_title  You So Crazy (1994)  Young Frankenstein (1974)  \
uid
1                           False                      True
2                           False                     False
3                           False                     False
4                           False                     False
5                           False                      True

movie_title  Young Guns (1988)  Young Guns II (1990)  \
uid
1                          True                 False
2                         False                 False
3                         False                 False
4                         False                 False
5                         False                 False

movie_title  Young Poisoner's Handbook, The (1995)  Zeus and Roxanne (1997)  \
uid
1                                            False                    False
2                                            False                    False
3                                            False                    False
4                                            False                    False
5                                            False                    False

movie_title  unknown  Á köldum klaka (Cold Fever) (1994)
uid
1               True                               False
2              False                               False
3              False                               False
4              False                               False
5               True                               False

[5 rows x 1664 columns]
```

0.0.3 Obtendo os itemsets mais frequntes

```
In [7]: frequent_itemsets = apriori(df_pivoted["id"], min_support=0.4, use_colnames=True, max_l

In [8]: frequent_itemsets

Out[8]:      support                                itemsets
        0   0.457052          (Air Force One (1997))
```

```
1   0.539767                                  (Contact (1997))
2   0.510074                    (English Patient, The (1996))
3   0.538706                                   (Fargo (1996))
4   0.437964                          (Godfather, The (1972))
5   0.454931                  (Independence Day (ID4) (1996))
6   0.407211                           (Jerry Maguire (1996))
7   0.514316                               (Liar Liar (1997))
8   0.417815                            (Pulp Fiction (1994))
9   0.445387                 (Raiders of the Lost Ark (1981))
10  0.537646                      (Return of the Jedi (1983))
11  0.400848                               (Rock, The (1996))
12  0.506893                                  (Scream (1996))
13  0.413574               (Silence of the Lambs, The (1991))
14  0.618240                               (Star Wars (1977))
15  0.479321                               (Toy Story (1995))
16  0.415695                          (Twelve Monkeys (1995))
17  0.417815                 (Star Wars (1977), Fargo (1996))
18  0.402969  (Star Wars (1977), Raiders of the Lost Ark (19...
19  0.509014      (Star Wars (1977), Return of the Jedi (1983))
20  0.404030               (Star Wars (1977), Toy Story (1995))
```

0.0.4 Obtendo as regras de associação

```
In [9]: rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.1)
```

0.0.5 Varredadura dos parâmetros

Para ajudar no chute dos parâmetros vou fazer uma varredura pra achar a ordem de grandeza das regras

```
In [10]: def logGridSearch(support_ths, confidence_ths, dataset):
             for th_it in support_ths:
         #         calculate frequent itemsets
                 frequent_itemsets = apriori(dataset, min_support=th_it, use_colnames=True, ma
                 num_itemsets = len(frequent_itemsets["itemsets"])
         #         generate rules
                 for th_rl in confidence_ths:
                     rules = []
                     if num_itemsets > 0:
                         rules = association_rules(frequent_itemsets, metric="confidence", min_
                     num_rules = len(rules)
                     print("( support_min: %f, confidence_min: %f ) #rules =  %i" % (th_it, th_
```

0.0.6 No. de regras x (min support, min cnfindence)

Para ter uma idea de como o número de regras varia de acordo com os parâmtros de min suport e condifence, vou executar uma varredura em escala log (estou interessado em orde de grandeza). O que depois vai ajudar a estabeleceros valores razoáveis para esses parâmetros

```
In [52]: th_min = np.log10(.05)
         th_max = np.log10(.2)
         ths_sup = np.logspace(th_min,th_max,num=10)

         th_min = np.log10(.7)
         th_max = np.log10(1)
         ths_cnf = np.logspace(th_min,th_max,num=10)

         logGridSearch(ths_sup, ths_cnf, df_pivoted["id"])
```

```
( support_min: 0.050000, confidence_min: 0.700000 ) #rules =  8846
( support_min: 0.050000, confidence_min: 0.728298 ) #rules =  6959
( support_min: 0.050000, confidence_min: 0.757741 ) #rules =  5247
( support_min: 0.050000, confidence_min: 0.788374 ) #rules =  3760
( support_min: 0.050000, confidence_min: 0.820245 ) #rules =  2497
( support_min: 0.050000, confidence_min: 0.853404 ) #rules =  1572
( support_min: 0.050000, confidence_min: 0.887904 ) #rules =  829
( support_min: 0.050000, confidence_min: 0.923799 ) #rules =  293
( support_min: 0.050000, confidence_min: 0.961144 ) #rules =  35
( support_min: 0.050000, confidence_min: 1.000000 ) #rules =  2
( support_min: 0.058326, confidence_min: 0.700000 ) #rules =  7296
( support_min: 0.058326, confidence_min: 0.728298 ) #rules =  5708
( support_min: 0.058326, confidence_min: 0.757741 ) #rules =  4350
( support_min: 0.058326, confidence_min: 0.788374 ) #rules =  3127
( support_min: 0.058326, confidence_min: 0.820245 ) #rules =  2113
( support_min: 0.058326, confidence_min: 0.853404 ) #rules =  1335
( support_min: 0.058326, confidence_min: 0.887904 ) #rules =  684
( support_min: 0.058326, confidence_min: 0.923799 ) #rules =  225
( support_min: 0.058326, confidence_min: 0.961144 ) #rules =  20
( support_min: 0.058326, confidence_min: 1.000000 ) #rules =  1
( support_min: 0.068040, confidence_min: 0.700000 ) #rules =  6086
( support_min: 0.068040, confidence_min: 0.728298 ) #rules =  4787
( support_min: 0.068040, confidence_min: 0.757741 ) #rules =  3684
( support_min: 0.068040, confidence_min: 0.788374 ) #rules =  2639
( support_min: 0.068040, confidence_min: 0.820245 ) #rules =  1756
( support_min: 0.068040, confidence_min: 0.853404 ) #rules =  1095
( support_min: 0.068040, confidence_min: 0.887904 ) #rules =  550
( support_min: 0.068040, confidence_min: 0.923799 ) #rules =  172
( support_min: 0.068040, confidence_min: 0.961144 ) #rules =  11
( support_min: 0.068040, confidence_min: 1.000000 ) #rules =  1
( support_min: 0.079370, confidence_min: 0.700000 ) #rules =  4897
( support_min: 0.079370, confidence_min: 0.728298 ) #rules =  3905
( support_min: 0.079370, confidence_min: 0.757741 ) #rules =  3025
( support_min: 0.079370, confidence_min: 0.788374 ) #rules =  2183
( support_min: 0.079370, confidence_min: 0.820245 ) #rules =  1478
( support_min: 0.079370, confidence_min: 0.853404 ) #rules =  930
( support_min: 0.079370, confidence_min: 0.887904 ) #rules =  455
( support_min: 0.079370, confidence_min: 0.923799 ) #rules =  144
```

```
( support_min: 0.079370, confidence_min: 0.961144 ) #rules =  9
( support_min: 0.079370, confidence_min: 1.000000 ) #rules =  1
( support_min: 0.092587, confidence_min: 0.700000 ) #rules =  3845
( support_min: 0.092587, confidence_min: 0.728298 ) #rules =  3069
( support_min: 0.092587, confidence_min: 0.757741 ) #rules =  2344
( support_min: 0.092587, confidence_min: 0.788374 ) #rules =  1660
( support_min: 0.092587, confidence_min: 0.820245 ) #rules =  1106
( support_min: 0.092587, confidence_min: 0.853404 ) #rules =  695
( support_min: 0.092587, confidence_min: 0.887904 ) #rules =  340
( support_min: 0.092587, confidence_min: 0.923799 ) #rules =  97
( support_min: 0.092587, confidence_min: 0.961144 ) #rules =  4
( support_min: 0.092587, confidence_min: 1.000000 ) #rules =  0
( support_min: 0.108006, confidence_min: 0.700000 ) #rules =  2927
( support_min: 0.108006, confidence_min: 0.728298 ) #rules =  2334
( support_min: 0.108006, confidence_min: 0.757741 ) #rules =  1804
( support_min: 0.108006, confidence_min: 0.788374 ) #rules =  1321
( support_min: 0.108006, confidence_min: 0.820245 ) #rules =  880
( support_min: 0.108006, confidence_min: 0.853404 ) #rules =  544
( support_min: 0.108006, confidence_min: 0.887904 ) #rules =  262
( support_min: 0.108006, confidence_min: 0.923799 ) #rules =  71
( support_min: 0.108006, confidence_min: 0.961144 ) #rules =  3
( support_min: 0.108006, confidence_min: 1.000000 ) #rules =  0
( support_min: 0.125992, confidence_min: 0.700000 ) #rules =  2196
( support_min: 0.125992, confidence_min: 0.728298 ) #rules =  1752
( support_min: 0.125992, confidence_min: 0.757741 ) #rules =  1333
( support_min: 0.125992, confidence_min: 0.788374 ) #rules =  962
( support_min: 0.125992, confidence_min: 0.820245 ) #rules =  629
( support_min: 0.125992, confidence_min: 0.853404 ) #rules =  389
( support_min: 0.125992, confidence_min: 0.887904 ) #rules =  194
( support_min: 0.125992, confidence_min: 0.923799 ) #rules =  55
( support_min: 0.125992, confidence_min: 0.961144 ) #rules =  2
( support_min: 0.125992, confidence_min: 1.000000 ) #rules =  0
( support_min: 0.146973, confidence_min: 0.700000 ) #rules =  1472
( support_min: 0.146973, confidence_min: 0.728298 ) #rules =  1184
( support_min: 0.146973, confidence_min: 0.757741 ) #rules =  913
( support_min: 0.146973, confidence_min: 0.788374 ) #rules =  681
( support_min: 0.146973, confidence_min: 0.820245 ) #rules =  453
( support_min: 0.146973, confidence_min: 0.853404 ) #rules =  283
( support_min: 0.146973, confidence_min: 0.887904 ) #rules =  137
( support_min: 0.146973, confidence_min: 0.923799 ) #rules =  36
( support_min: 0.146973, confidence_min: 0.961144 ) #rules =  0
( support_min: 0.146973, confidence_min: 1.000000 ) #rules =  0
( support_min: 0.171449, confidence_min: 0.700000 ) #rules =  954
( support_min: 0.171449, confidence_min: 0.728298 ) #rules =  781
( support_min: 0.171449, confidence_min: 0.757741 ) #rules =  601
( support_min: 0.171449, confidence_min: 0.788374 ) #rules =  435
( support_min: 0.171449, confidence_min: 0.820245 ) #rules =  279
( support_min: 0.171449, confidence_min: 0.853404 ) #rules =  177
```

```
( support_min: 0.171449, confidence_min: 0.887904 ) #rules =  83
( support_min: 0.171449, confidence_min: 0.923799 ) #rules =  24
( support_min: 0.171449, confidence_min: 0.961144 ) #rules =  0
( support_min: 0.171449, confidence_min: 1.000000 ) #rules =  0
( support_min: 0.200000, confidence_min: 0.700000 ) #rules =  543
( support_min: 0.200000, confidence_min: 0.728298 ) #rules =  457
( support_min: 0.200000, confidence_min: 0.757741 ) #rules =  362
( support_min: 0.200000, confidence_min: 0.788374 ) #rules =  265
( support_min: 0.200000, confidence_min: 0.820245 ) #rules =  175
( support_min: 0.200000, confidence_min: 0.853404 ) #rules =  115
( support_min: 0.200000, confidence_min: 0.887904 ) #rules =  49
( support_min: 0.200000, confidence_min: 0.923799 ) #rules =  14
( support_min: 0.200000, confidence_min: 0.961144 ) #rules =  0
( support_min: 0.200000, confidence_min: 1.000000 ) #rules =  0
```

```
In [12]: frequent_itemsets = apriori(df_pivoted["id"], min_support=0.1, use_colnames=True, max
```

```
In [13]: rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.92)
```

```
In [14]: rules.sort_values(["antecedent support", "confidence"]).head()
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
58	(Miracle on 34th Street (1994))	(Star Wars (1977))	0.107105	0.618240	0.100742	0.940594	1.521407	0.034526	6.426299
64	(Pinocchio (1940))	(Star Wars (1977))	0.107105	0.618240	0.100742	0.940594	1.521407	0.034526	6.426299
95	(Young Guns (1988))	(Star Wars (1977))	0.107105	0.618240	0.100742	0.940594	1.521407	0.034526	6.426299
73	(True Romance (1993))	(Raiders of the Lost Ark (1981))	0.110286	0.445387	0.102863	0.932692	2.094116	0.053743	8.239964
4	(Outbreak (1995))	(Back to the Future (1985))	0.110286	0.371156	0.103924	0.942308	2.538846	0.062990	10.899965

Como esperado, quando trabalhamos com valores de suporte biaxo, uma regra com valor de condiaça alta pode não ser uma regra forte.

```
In [15]: print("Valor médio do lift para as 100 regras geradas com supoort min %f r confidence
```

```
Valor médio do lift para as 100 regras geradas com supoort min 0.100000 r confidence min 0.920
```

A partir desses expermientos, resolvi gerar regras fortes com * **suporte mínimo 10%** * **confiança mínima de 75%%** * **lift mínimo de 3**

```
In [16]: rules2 = association_rules(frequent_itemsets, metric="confidence", min_threshold=.75)
```

```
In [17]: print("lift médio das regras frequentes gerados com suporte mínimo fixo: %f " % rules1
```

```
lift médio das regras frequentes gerados com suporte mínimo fixo: 2.039551
```

```
In [18]: print("quantidade de regras atendendo o critério de confiança mínima estabelecidoo: %
```

```
quantidade de regras atendendo o critério de confiança mínima estabelecidoo: 2205
```

```
In [19]: rules_strong = rules2[rules2['lift'] > 3]
```

```
In [20]: print("quantidade de regras atendendo o critério de confiança e de ift mínimos estabel
```

```
quantidade de regras atendendo o critério de confiança e de ift mínimos estabelecidoo: 98
```

```
In [21]: rules_strong.head()
```

```
Out[21]:                        antecedents              consequents  \
         48  (Beauty and the Beast (1991))         (Aladdin (1992))
         49             (Cinderella (1950))         (Aladdin (1992))
         50                  (Dumbo (1941))         (Aladdin (1992))
         55                 (Aladdin (1992))  (Lion King, The (1994))
         56         (Lion King, The (1994))         (Aladdin (1992))

             antecedent support  consequent support   support  confidence      lift  \
         48            0.214210            0.232238  0.163309    0.762376  3.282743
         49            0.136797            0.232238  0.103924    0.759690  3.271176
         50            0.130435            0.232238  0.100742    0.772358  3.325723
         55            0.232238            0.233298  0.177094    0.762557  3.268597
         56            0.233298            0.232238  0.177094    0.759091  3.268597

             leverage  conviction
         48  0.113561    3.231000
         49  0.072154    3.194882
         50  0.070450    3.372671
         55  0.122914    3.228995
         56  0.122914    3.186939
```

0.0.7 calculando o chisquare para essas regras

Podemos utilizar uma função do pacote scipy, mas para isso, para cada linha do dataset de rules, teremos que calcular a tabela de contigênca. A documentação da função tem o seguinte exemlo:

```
obs = np.array([[10, 10, 20], [20, 20, 20]])
chi2_contingency(obs)
```

```
In [22]: def selectComplementByFilm(dataset, film):
             wihFilm = dataset[dataset[film] == True]
             notWihFilm = dataset[dataset[film] == False]
             return wihFilm, notWihFilm
```

```
In [23]: def createContigencyTable(dataset, filmA, filmB):
             dfA, dfnA = selectComplementByFilm(dataset,filmA)
             dfAB, dfAnB = selectComplementByFilm(dfA, filmB)
             dfnAB, dfnAnB = selectComplementByFilm(dfnA, filmB)
             return np.array([[len(dfAB), len(dfAnB)], [len(dfnAB), len(dfnAnB)]])
```

```
In [24]: def calculateChiSquareForFule(df_pivoted, r):
             ant = str(r["antecedents"])
             ant = re.search("\\{'(.*)'\\}", ant)[1]
             des = str(r["consequents"])
             des = re.search("\\{'(.*)'\\}", des)[1]
             table = createContigencyTable(df_pivoted, ant, des)
             pvalue = chi2_contingency(table)[1]
             return pvalue
```

```
In [25]: rules_strong['chi2-pvalue'] = rules_strong.apply(lambda r: calculateChiSquareForFule(
         # df_pivoted["id"].apply(lambda x: x, axis=1)
```

```
C:\Users\souza\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
  """Entry point for launching an IPython kernel.
```

```
In [26]: print("Números de regras que falham ao teste de independência com confiança .001", (ru
```

```
Números de regras que falham ao teste de independência com confiança .001 0
```

0.0.8 Export data to .csv

```
In [27]: def createOutFileRuleFromat(r):
             ant = str(r["antecedents"])
             ant = re.search("\\{'(.*)'\\}", ant)[1]
             des = str(r["consequents"])
             des = re.search("\\{'(.*)'\\}", des)[1]
             return ant + ' => ' + des
```

```
In [28]: def exportRulesToTxt(rules):
         #     Regra | suporte | confiança | lift (B,C) e (B,!C) | chi-square...
             out = pd.DataFrame()
             out["regra"] = rules.apply(createOutFileRuleFromat, axis=1)
             out['suporte'] = rules['support']
             out['confiança'] = rules['confidence']
             out['lift (A=>B)'] = rules['lift']
             out['chi-square (pvalue)'] = rules['chi2-pvalue']
             out = out.sort_values(["suporte", "confiança", 'lift (A=>B)'])
             return out

In [29]: out = exportRulesToTxt(rules_strong)

In [30]: out.head()
```

Out[30]:

	regra	suporte	confiança	\
50	Dumbo (1941) => Aladdin (1992)	0.100742	0.772358	
826	Dumbo (1941) => Snow White and the Seven Dwarf...	0.100742	0.772358	
335	Batman Forever (1995) => Speed (1994)	0.100742	0.833333	
338	Batman Forever (1995) => Top Gun (1986)	0.100742	0.833333	
2112	Stargate (1994) => Star Trek III: The Search f...	0.101803	0.755906	

	lift (A=>B)	chi-square (pvalue)
50	3.325723	1.658784e-51
826	4.234496	8.731407e-73
335	3.416667	2.790322e-54
338	3.571970	6.924089e-58
2112	4.168532	5.529445e-72

```
In [31]: out.to_csv(r'..\ProcessedData\rules100k.csv', index = None, header=True) #Don't forge
```

0.0.9 Executando análise para a base de 1M ratings

```
In [32]: ratingsDF_1m = pd.read_pickle("..\\ProcessedData\\df_1m.pkl")

In [33]: ratingsDF_1m.head(5)
```

Out[33]:

	uid	id	rating	timestamp	\
0	1	1193	5	2000-12-31 22:12:40	
1	2	1193	5	2000-12-31 21:33:33	
2	12	1193	4	2000-12-30 23:49:39	
3	15	1193	4	2000-12-30 18:01:19	
4	17	1193	5	2000-12-30 06:41:11	

	movie_title	Genres
0	ONE FLEW OVER THE CUCKOO'S NEST (1975)	Drama
1	ONE FLEW OVER THE CUCKOO'S NEST (1975)	Drama
2	ONE FLEW OVER THE CUCKOO'S NEST (1975)	Drama
3	ONE FLEW OVER THE CUCKOO'S NEST (1975)	Drama
4	ONE FLEW OVER THE CUCKOO'S NEST (1975)	Drama

In [34]:
```
df_pivoted_1m = pd.pivot_table(ratingsDF_1m, columns=['movie_title'], values=['id'], i
df_pivoted_1m.fillna(False, inplace=True)
```

In [35]:
```
df_pivoted_1m.head(10)
```

Out[35]:

	id			\
movie_title	$1,000,000 DUCK (1971)	'BURBS, THE (1989)	'NIGHT MOTHER (1986)	
uid				
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	
5	False	False	False	
6	False	False	False	
7	False	False	False	
8	False	False	False	
9	False	False	False	
10	False	True	False	

		\
movie_title	'TIL THERE WAS YOU (1997)	...AND JUSTICE FOR ALL (1979)
uid		
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False

		\
movie_title	1-900 (1994)	10 THINGS I HATE ABOUT YOU (1999)
uid		
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False

\

```
movie_title 101 DALMATIANS (1961) 101 DALMATIANS (1996) 12 ANGRY MEN (1957)
uid
1                           False                 False                False
2                           False                 False                False
3                           False                 False                False
4                           False                 False                False
5                           False                 False                False
6                           False                 False                False
7                           False                 False                False
8                           False                 False                False
9                           False                 False                False
10                          False                 False                 True

                      ...                                                    \
movie_title           ...           YOUNG GUNS (1988) YOUNG GUNS II (1990)
uid                   ...
1                     ...                       False                False
2                     ...                       False                False
3                     ...                        True                 True
4                     ...                       False                False
5                     ...                       False                False
6                     ...                       False                False
7                     ...                       False                False
8                     ...                       False                False
9                     ...                       False                False
10                    ...                       False                False

                                                 \
movie_title YOUNG POISONER'S HANDBOOK, THE (1995)
uid
1                                          False
2                                          False
3                                          False
4                                          False
5                                          False
6                                          False
7                                          False
8                                          False
9                                          False
10                                         False

                                                                          \
movie_title YOUNG SHERLOCK HOLMES (1985) YOUR FRIENDS AND NEIGHBORS (1998)
uid
1                                  False                              False
2                                  False                              False
3                                  False                              False
4                                  False                              False
```

```
5                                   False                                False
6                                   False                                False
7                                   False                                False
8                                   False                                False
9                                   False                                False
10                                  False                                False

                                                                                \
movie_title ZACHARIAH (1971) ZED & TWO NOUGHTS, A (1985) ZERO EFFECT (1998)
uid
1                        False                        False             False
2                        False                        False             False
3                        False                        False             False
4                        False                        False             False
5                        False                        False             False
6                        False                        False             False
7                        False                        False             False
8                        False                        False             False
9                        False                        False             False
10                       False                        False             False


movie_title ZERO KELVIN (KJÆRLIGHETENS KJØTERE) (1995) ZEUS AND ROXANNE (1997)
uid
1                                                  False                  False
2                                                  False                  False
3                                                  False                  False
4                                                  False                  False
5                                                  False                  False
6                                                  False                  False
7                                                  False                  False
8                                                  False                  False
9                                                  False                  False
10                                                 False                  False

[10 rows x 3706 columns]
```

```
In [36]: frequent_itemsets_1m = apriori(df_pivoted_1m["id"], min_support=0.1, use_colnames=True
```

```
In [37]: rules_1m = association_rules(frequent_itemsets_1m, metric="confidence", min_threshold=
```

```
In [38]: rules_1m
```

```
Out[38]:                              antecedents  \
0      (BACK TO THE FUTURE PART II (1989))
1                             (DUNE (1984))
2                       (HIGHLANDER (1986))
3                  (RUNNING MAN, THE (1987))
4   (STAR TREK IV: THE VOYAGE HOME (1986))
```

```
5   (STAR TREK: THE MOTION PICTURE (1979))
6   (STAR TREK: THE MOTION PICTURE (1979))
7                        (SUPERMAN (1978))
8                     (SUPERMAN II (1980))
9                            (TRON (1982))
10                    (SUPERMAN II (1980))
11                           (TRON (1982))

                                          consequents  antecedent support  \
0                          (BACK TO THE FUTURE (1985))            0.191722
1         (STAR WARS: EPISODE IV - A NEW HOPE (1977))            0.130629
2   (STAR WARS: EPISODE V - THE EMPIRE STRIKES BAC...            0.122682
3   (STAR WARS: EPISODE V - THE EMPIRE STRIKES BAC...            0.120861
4   (STAR WARS: EPISODE V - THE EMPIRE STRIKES BAC...            0.186424
5         (STAR WARS: EPISODE IV - A NEW HOPE (1977))            0.131126
6   (STAR WARS: EPISODE V - THE EMPIRE STRIKES BAC...            0.131126
7         (STAR WARS: EPISODE IV - A NEW HOPE (1977))            0.202318
8         (STAR WARS: EPISODE IV - A NEW HOPE (1977))            0.140066
9         (STAR WARS: EPISODE IV - A NEW HOPE (1977))            0.160596
10  (STAR WARS: EPISODE V - THE EMPIRE STRIKES BAC...            0.140066
11  (STAR WARS: EPISODE V - THE EMPIRE STRIKES BAC...            0.160596

    consequent support   support  confidence      lift  leverage  conviction
0             0.427649  0.176987    0.923143  2.158647  0.094997    7.446994
1             0.495199  0.120364    0.921420  1.860707  0.055677    6.424004
2             0.495033  0.113742    0.927126  1.872856  0.053010    6.929268
3             0.495033  0.113079    0.935616  1.890008  0.053249    7.843103
4             0.495033  0.171854    0.921847  1.862193  0.079568    6.461281
5             0.495199  0.123675    0.943182  1.904653  0.058742    8.884503
6             0.495033  0.121192    0.924242  1.867032  0.056280    6.665563
7             0.495199  0.190894    0.943535  1.905367  0.090706    8.940105
8             0.495199  0.128974    0.920804  1.859463  0.059613    6.374059
9             0.495199  0.147848    0.920619  1.859089  0.068321    6.359186
10            0.495033  0.131291    0.937352  1.893514  0.061954    8.060415
11            0.495033  0.148179    0.922680  1.863876  0.068678    6.530905
```

```
In [39]: print("quantidade de regras atendendo o critério de confiança mínima estabelecidoo: %
```

```
quantidade de regras atendendo o critério de confiança mínima estabelecidoo: 12
```

```
In [40]: strong_rules_1m = rules_1m[rules_1m['lift'] > 3]
```

```
In [41]: print("quantidade de regras atendendo o critério de confiança e de ift mínimos estabel
```

```
quantidade de regras atendendo o critério de confiança e de ift mínimos estabelecidoo: 0
```

Teste com suporte mínimo igual a 10.000(absoluto)

```
In [42]: frequent_itemsets_1m2 = apriori(df_pivoted_1m["id"], min_support=0.01, use_colnames=Tr

In [43]: rules_1m2 = association_rules(frequent_itemsets_1m2, metric="confidence", min_thresho

In [44]: rules_1m2.head()
```

Out[44]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ALPHAVILLE (1965))	(2001: A SPACE ODYSSEY (1968))	0.010596	0.284106	0.010099	0.953125	3.354822	0.007089	15.272406
1	(QUATERMASS AND THE PIT (1967))	(2001: A SPACE ODYSSEY (1968))	0.011424	0.284106	0.010596	0.927536	3.264755	0.007350	9.879338
2	(JURY DUTY (1995))	(ACE VENTURA: PET DETECTIVE (1994))	0.013411	0.126821	0.012417	0.925926	7.301035	0.010716	11.787914
3	(AFFLICTION (1997))	(AMERICAN BEAUTY (1999))	0.032781	0.567550	0.030629	0.934343	1.646276	0.012024	6.586551
4	(YOUNG DOCTORS IN LOVE (1982))	(AIRPLANE! (1980))	0.013079	0.286589	0.012086	0.924051	3.224301	0.008338	9.393240

```
In [45]: print("quantidade de regras atendendo o critério de confiança mínima estabelecidoo: %

quantidade de regras atendendo o critério de confiança mínima estabelecidoo: 237

In [46]: print("lift médio das regras frequentes gerados com suporte mínimo fixo: %f " % rules_

lift médio das regras frequentes gerados com suporte mínimo fixo: 2.263238

In [47]: strong_rules_1m2 = rules_1m2[rules_1m2['lift'] > 3]

In [48]: print("quantidade de regras atendendo o critério de confiança e de ift mínimos estabel

quantidade de regras atendendo o critério de confiança e de ift mínimos estabelecidoo: 18

In [49]: strong_rules_1m2
```

Out[49]:

	antecedents
0	(ALPHAVILLE (1965))
1	(QUATERMASS AND THE PIT (1967))
2	(JURY DUTY (1995))
4	(YOUNG DOCTORS IN LOVE (1982))
5	(ALL DOGS GO TO HEAVEN 2 (1996))
6	(RETURN OF JAFAR, THE (1993))
52	(DEATH WISH II (1982))
62	(EXORCIST II: THE HERETIC (1977))
77	(MAX DUGAN RETURNS (1983))
96	(SOLO (1996))
97	(SPECIALIST, THE (1994))
98	(UNIVERSAL SOLDIER: THE RETURN (1999))
106	(JAWS 2 (1978))
196	(UNDER THE RAINBOW (1981))
197	(YOUNG DOCTORS IN LOVE (1982))
198	(ROMEO IS BLEEDING (1993))
206	(SOLO (1996))
208	(SPECIALIST, THE (1994))

	consequents	antecedent support
0	(2001: A SPACE ODYSSEY (1968))	0.010596
1	(2001: A SPACE ODYSSEY (1968))	0.011424
2	(ACE VENTURA: PET DETECTIVE (1994))	0.013411
4	(AIRPLANE! (1980))	0.013079
5	(ALADDIN (1992))	0.012417
6	(ALADDIN (1992))	0.026987
52	(LETHAL WEAPON (1987))	0.014073
62	(EXORCIST, THE (1973))	0.018709
77	(FERRIS BUELLER'S DAY OFF (1986))	0.015894
96	(INDEPENDENCE DAY (ID4) (1996))	0.017550
97	(INDEPENDENCE DAY (ID4) (1996))	0.025331
98	(INDEPENDENCE DAY (ID4) (1996))	0.015066
106	(JAWS (1975))	0.061258
196	(ROMANCING THE STONE (1984))	0.011755
197	(ROMANCING THE STONE (1984))	0.013079
198	(USUAL SUSPECTS, THE (1995))	0.019371
206	(STARGATE (1994))	0.017550
208	(SPEED (1994))	0.025331

	consequent support	support	confidence	lift	leverage	conviction
0	0.284106	0.010099	0.953125	3.354822	0.007089	15.272406
1	0.284106	0.010596	0.927536	3.264755	0.007350	9.879338
2	0.126821	0.012417	0.925926	7.301035	0.010716	11.787914
4	0.286589	0.012086	0.924051	3.224301	0.008338	9.393240
5	0.223675	0.011589	0.933333	4.172712	0.008812	11.644868
6	0.223675	0.024834	0.920245	4.114199	0.018798	9.733915
52	0.269371	0.013079	0.929412	3.450306	0.009289	10.350579

62	0.146523	0.017219	0.920354	6.281286	0.014477	10.715876
77	0.243874	0.014901	0.937500	3.844196	0.011025	12.098013
96	0.286424	0.016391	0.933962	3.260770	0.011364	10.805582
97	0.286424	0.023344	0.921569	3.217500	0.016089	9.098096
98	0.286424	0.013907	0.923077	3.222766	0.009592	9.276490
106	0.280960	0.056623	0.924324	3.289876	0.039411	9.501596
196	0.222682	0.011093	0.943662	4.237709	0.008475	13.797392
197	0.222682	0.012086	0.924051	4.149640	0.009174	10.234685
198	0.295199	0.018377	0.948718	3.213829	0.012659	13.743626
206	0.184768	0.016225	0.924528	5.003719	0.012983	10.801821
208	0.273179	0.023344	0.921569	3.373500	0.016424	9.266970

0.0.10 Conclusões

O algoritmo utilizado O processo de seleção de regras fortes utilizado consistiu na execução de 3 passos:

1. Encontrar todos os itemsets de cumprimento 2 com suporte mínimo de 10%
2. Dado o conjunto anterior, achar todas as regras de associação com confiança minima de 75%
3. filtrar as regras e selecionar como regras fortes apenas as que obtiveram lift superior a 3

A seleção poderia ter sido feita levando em consideração apenas o lift e ignorando a confiança, mas preferi fazer a seleção de regras em dois passos porque caso o objetivo fosse realmente construir um sistema de recomendação de filmes para usuários, uma regra de lift alto, mas confiança baixa não configuraria uma boa recomendação. Por outro lado, uma regra com confiança alta, mas lift baixo, pode representar uma recomendação óbvia, já que tende a ser uma recomendação de blockbuster e outros filmes populares que o usuário poderia já ter em mente.

Discussão sobre as regras geradas No conjunto de 100K ratings, o limiar de 10% de suporte ajudou a filtrar "filmes de nicho" (s pouco frequentes na base), mas sem enviesar a análise apenas para os filmes blockbuster(os de grande sucesso de público). O motivo de eu tentar preservar alguns filmes não blockbuster entre os itemsets considerados frequentes, é que eu percebi através de exploração manual, que os blockbusters geram regras de confiança alta o que configuram recomendações óbvias, o que vem do fato de quase todo mundo assisti-los independentemente de nuances de gosto.

O resultado da aplicação dos limiares for permitiu a geração uma lista de quase 100 regras com lifts altos, para uma confiança mínima intermediária (75%). Acho que isso apresentaria aos usuários de um sistema de recomendação filmes inesperados por vezes, o que poderia surpreendê-los positivamente. Todas as regras rejeitam a hipótese nula do teste chi-quadrado com confiança acima de 99%

A aplicação das mesmas regras no dataset de 1M ratings no entanto, não gerou regras relevantes na mesma quantidade que foram geradas no dataset de 100k, mesmo para o caso em que considerei o valor absoluto do suporte mínimo e não seu valor proporcional. Isso, acredito, deve-se ao fato da base de 100k não ser mera amostra da base de 1M. A base de 1M apresenta filmes que não estão presentes na de 100k, (16 3883 títulos na primeira e 1664 na segunda), O mesmo deve acontecer para os usuários e portanto as propriedades estatística de uma não são representantes da outra.