

Loading functions and modules

```
In [ ]: import pandas as pd
import numpy as np
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

Loading Data

```
In [ ]: df = pd.read_csv("../data/final_0_80509.csv")
df.head()
```

C:\Users\user\AppData\Local\Temp\ipykernel_12092\1952620361.py:1: DtypeWarning: Columns (10,35) have mixed types. Specify dtype option on import or set low_memory=False.
df = pd.read_csv("../data/final_0_80509.csv")

Out[]:

	Index	AppID	Title	Initial_Price	Final_Price	Discount_Percent	Developers	Publishers	Genres	Categories	...	Subtitle_1
0	0	20200	Galactic Bowling	NaN	NaN	NaN	['Perpetual FX Creative']	['Perpetual FX Creative']	['Casual', 'Indie', 'Sports']	['Single-player', 'Multi-player', 'Steam Achievements']	...	
1	1	655370	Train Bandit	52.0	52.0	0.0	['Rusty Moyher']	['Wild Rooster']	['Action', 'Indie']	['Single-player', 'Steam Achievements', 'Full ...']	...	['English', 'Italian']
2	2	1732930	Jolt Project	199.0	199.0	0.0	['Campião Games']	['Campião Games']	['Action', 'Adventure', 'Indie', 'Strategy']	['Single-player']	...	
3	3	1355720	Henosis™	NaN	NaN	NaN	['Odd Critter Games']	['Odd Critter Games']	['Adventure', 'Casual', 'Indie']	['Single-player', 'Full controller support']	...	
4	4	1139950	Two Weeks in Painland	0.0	0.0	0.0	['Unusual Games']	['Unusual Games']	['Adventure', 'Indie']	['Single-player', 'Steam Achievements']	...	['English']

5 rows × 37 columns

Looking at all genres

```
In [ ]: n = 80509
dataset = []
for genres in df["Genres"][:n]:
    if genres is not np.nan:
        dataset.append(eval(genres))

dataset[:10]
```

Out[]:

```
[['Casual', 'Indie', 'Sports'],
 ['Action', 'Indie'],
 ['Action', 'Adventure', 'Indie', 'Strategy'],
 ['Adventure', 'Casual', 'Indie'],
 ['Adventure', 'Indie'],
 ['Adventure',
 'Casual',
 'Free to Play',
 'Massively Multiplayer',
 'RPG',
 'Strategy'],
 ['Indie', 'Strategy'],
 ['Casual'],
 ['Adventure', 'RPG', 'Simulation', 'Strategy'],
 ['Action', 'Adventure', 'Indie']]
```

Converting genres to asymmetric binary attributes

```
In [ ]: te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df2 = pd.DataFrame(te_ary, columns=te.columns_)
df2
```

Out[]:

	360 Video	Accounting	Action	Adventure	Animation & Modeling	Audio Production	Casual	Design & Illustration	Documentary	Early Access	...	Short	Simulation
0	False	False	False	False	False	False	True	False	False	False	...	False	False
1	False	False	True	False	False	False	False	False	False	False	...	False	False
2	False	False	True	True	False	False	False	False	False	False	...	False	False
3	False	False	False	True	False	False	True	False	False	False	...	False	False
4	False	False	False	True	False	False	False	False	False	False	...	False	False
...
75616	False	False	False	False	False	False	False	False	False	False	...	False	False
75617	False	False	True	False	False	False	True	False	False	False	...	False	True
75618	False	False	False	False	False	False	False	False	False	False	...	False	True
75619	False	False	False	True	False	False	False	False	False	False	...	False	False
75620	False	False	True	False	False	False	False	False	False	False	...	False	False

75621 rows × 33 columns



Rule minning

min support 50%, confidence 75%

In []:

```
frq_items = apriori(df2, min_support=0.2, use_colnames=True)
rules = association_rules(frq_items, min_threshold = .4, metric="confidence")
rules.sort_values(["support"],ascending=False)
```

Out[]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
4	(Casual)	(Indie)	0.407109	0.688407	0.310165	0.761872	1.106718	0.029909	1.308514	0.162640
5	(Indie)	(Casual)	0.688407	0.407109	0.310165	0.450555	1.106718	0.029909	1.079073	0.309467
0	(Indie)	(Action)	0.688407	0.419540	0.310020	0.450344	1.073424	0.021206	1.056043	0.219522
1	(Action)	(Indie)	0.419540	0.688407	0.310020	0.738952	1.073424	0.021206	1.193626	0.117840
2	(Adventure)	(Indie)	0.383941	0.688407	0.290316	0.756148	1.098403	0.026009	1.277797	0.145420
3	(Indie)	(Adventure)	0.688407	0.383941	0.290316	0.421722	1.098403	0.026009	1.065334	0.287514

Conclusion of good rules

- 1. Casual <-> Indie . Most Casual games are made by an individual and vice versa
- 2. Even on basis of Action <-> Indie . Most Actions games are made by an individual and vice versa
- 3. And on basis of Adventure <-> Indie . Most Adventure games are made by an individual and vice versa