References:

https://github.com/ymoch/apyori

https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/

https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html
**Installing Apyori is a simple implementation of Apriori algorithm with Python 2.7 and 3.3 - 3.5, provided as APIs and as commandline interfaces.

Supports a JSON output format. Supports a TSV output format for 2-items relations. **

Double-click (or enter) to edit

```
1 !pip install apyori
```

```
Requirement already satisfied: apyori in /usr/local/lib/python3.6/dist-packages (1.1
```

Importing Libraries

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 from apyori import apriori
```

Reading data into data frame

```
1 store_data = pd.read_csv('store_data.csv')
```

```
1 store_data = pd.read_csv('store_data.csv', header=None)
```

Displaying the data

```
1 store_data.head()
```

	0	1	2	3	4	5	6	7	8	9
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Data Preprocessing The Apriori library we are going to use requires our dataset to be in the form of a list of lists, where the whole dataset is a big list and each transaction in the dataset is an inner list within the outer big list. To convert pandas dataframe into a list of lists, execute the following script

```
records = []
for i in range(0, 7501):
    records.append([str(store_data.values[i,j]) for j in range(0, 20)])


```

Applying Apriori The next step is to apply the **Apriori algorithm** on the dataset. To do so, we can use the apriori class that we imported from the apyori library.

The **apriori class** requires some parameter values to work. The first parameter is the list of list that you want to extract rules from. The second parameter is the min_support parameter. This parameter is used to select the items with support values greater than the value specified by the parameter. Next, the min_confidence parameter filters those rules that have confidence greater than the confidence threshold specified by the parameter. Similarly, the min_lift parameter specifies the minimum lift value for the short listed rules. Finally, the min_length parameter specifies the minimum number of items that you want in your rules.

Let's suppose that we want rules for only those items that are purchased at least 5 times a day, or 7 x 5 = 35 times in one week, since our dataset is for a one-week time period.

The support for those items can be calculated as

35/7500 = 0.0045.

The minimum confidence for the rules is 20% or 0.2.

Similarly, we specify the value for lift as 3 and finally min_length is 2 since we want at least two products in our rules.

These values are mostly just arbitrarily chosen, so you can play with these values and see what difference it makes in the rules you get back out.

```
association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3
association_results = list(association_rules)

for i in range(0, len(association_results)):
    print(association_results[i][0])
```

```
frozenset({'chicken', 'light cream'})
frozenset({'mushroom cream sauce', 'escalope'})
frozenset({'pasta', 'escalope'})
frozenset({'herb & pepper', 'ground beef'})
frozenset({'tomato sauce', 'ground beef'})
frozenset({'olive oil', 'whole wheat pasta'})
frozenset({'pasta', 'shrimp'})
frozenset({'chicken', 'nan', 'light cream'})
frozenset({'chocolate', 'shrimp', 'frozen vegetables'})
frozenset({'spaghetti', 'cooking oil', 'ground beef'})
frozenset({'mushroom cream sauce', 'nan', 'escalope'})
frozenset({'nan', 'pasta', 'escalope'})
frozenset({'spaghetti', 'ground beef', 'frozen vegetables'})
frozenset({'milk', 'olive oil', 'frozen vegetables'})
frozenset({'mineral water', 'shrimp', 'frozen vegetables'})
frozenset({'spaghetti', 'olive oil', 'frozen vegetables'})
frozenset({'spaghetti', 'shrimp', 'frozen vegetables'})
frozenset({'spaghetti', 'tomatoes', 'frozen vegetables'})
frozenset({'spaghetti', 'grated cheese', 'ground beef'})
frozenset({'herb & pepper', 'mineral water', 'ground beef'})
frozenset({'nan', 'herb & pepper', 'ground beef'})
frozenset({'spaghetti', 'herb & pepper', 'ground beef'})
frozenset({'milk', 'olive oil', 'ground beef'})
frozenset({'nan', 'tomato sauce', 'ground beef'})
frozenset({'spaghetti', 'shrimp', 'ground beef'})
frozenset({'spaghetti', 'milk', 'olive oil'})
frozenset({'soup', 'olive oil', 'mineral water'})
frozenset({'nan', 'olive oil', 'whole wheat pasta'})
frozenset({'nan', 'pasta', 'shrimp'})
frozenset({'spaghetti', 'olive oil', 'pancakes'})
frozenset({'nan', 'chocolate', 'shrimp', 'frozen vegetables'})
frozenset({'spaghetti', 'nan', 'cooking oil', 'ground beef'})
frozenset({'spaghetti', 'nan', 'ground beef', 'frozen vegetables'})
frozenset({'spaghetti', 'milk', 'mineral water', 'frozen vegetables'})
frozenset({'milk', 'nan', 'olive oil', 'frozen vegetables'})
frozenset({'nan', 'mineral water', 'shrimp', 'frozen vegetables'})
frozenset({'spaghetti', 'nan', 'olive oil', 'frozen vegetables'})
frozenset({'spaghetti', 'nan', 'shrimp', 'frozen vegetables'})
frozenset({'spaghetti', 'nan', 'tomatoes', 'frozen vegetables'})
frozenset({'spaghetti', 'nan', 'grated cheese', 'ground beef'})
frozenset({'nan', 'herb & pepper', 'mineral water', 'ground beef'})
frozenset({'spaghetti', 'nan', 'herb & pepper', 'ground beef'})
frozenset({'milk', 'nan', 'olive oil', 'ground beef'})
frozenset({'spaghetti', 'nan', 'shrimp', 'ground beef'})
```

Display the rule, the support, the confidence, and lift for each rule

```
for item in association_results:

    # first index of the inner list
    # Contains base item and add item
    pair = item[0]
    items = [x for x in pair]
    print("Rule: " + items[0] + " -> " + items[1])

```

```
    #second index of the inner list
    print("Support: " + str(item[1]))

    #third index of the list located at 0th
    #of the third index of the inner list

    print("Confidence: " + str(item[2][0][2]))
    print("Lift: " + str(item[2][0][3]))
    print("====================================")
```

```
Rule: chicken -> light cream
Support: 0.004532728969470737
Confidence: 0.29059829059829057
Lift: 4.84395061728395
====================================
Rule: mushroom cream sauce -> escalope
Support: 0.005732568990801226
Confidence: 0.3006993006993007
Lift: 3.790832696715049
====================================
Rule: pasta -> escalope
Support: 0.005865884548726837
Confidence: 0.3728813559322034
Lift: 4.700811850163794
====================================
Rule: herb & pepper -> ground beef
Support: 0.015997866951073192
Confidence: 0.3234501347708895
Lift: 3.2919938411349285
====================================
Rule: tomato sauce -> ground beef
Support: 0.005332622317024397
Confidence: 0.3773584905660377
Lift: 3.840659481324083
====================================
Rule: olive oil -> whole wheat pasta
Support: 0.007998933475536596
Confidence: 0.2714932126696833
Lift: 4.122410097642296
====================================
Rule: pasta -> shrimp
Support: 0.005065991201173177
Confidence: 0.3220338983050847
Lift: 4.506672147735896
====================================
Rule: chicken -> nan
Support: 0.004532728969470737
Confidence: 0.29059829059829057
Lift: 4.84395061728395
====================================
Rule: chocolate -> shrimp
Support: 0.005332622317024397
Confidence: 0.23255813953488375
Lift: 3.2545123221103784
====================================
Rule: spaghetti -> cooking oil
Support: 0.004799360085321957
Confidence: 0.5714285714285714
Lift: 3.2819951870487856
====================================
Rule: mushroom cream sauce -> nan
Support: 0.005732568990801226
Confidence: 0.3006993006993007
Lift: 3.790832696715049
====================================
Rule: nan -> pasta
Support: 0.005865884548726837
Confidence: 0.3728813559322034
Lift: 4.700811850163794
====================================
Rule: spaghetti -> ground beef
```

```
Support: 0.008665511265164644
Confidence: 0.31100478468899523
Lift: 3.165328208890303
====================================
Rule: milk -> olive oil
Support: 0.004799360085321957
Confidence: 0.20338983050847456
Lift: 3.088314005352364
====================================
Rule: mineral water -> shrimp
Support: 0.007199040127982935
Confidence: 0.30508474576271183
Lift: 3.200616332819722
====================================
Rule: spaghetti -> olive oil
Support: 0.005732568990801226
Confidence: 0.20574162679425836
Lift: 3.1240241752707125
====================================
Rule: spaghetti -> shrimp
Support: 0.005999200106652446
Confidence: 0.21531100478468898
Lift: 3.0131489680782684
====================================
Rule: spaghetti -> tomatoes
Support: 0.006665777896280496
Confidence: 0.23923444976076558
Lift: 3.4980460188216425
====================================
Rule: spaghetti -> grated cheese
Support: 0.005332622317024397
Confidence: 0.3225806451612903
Lift: 3.283144395325426
====================================
Rule: herb & pepper -> mineral water
Support: 0.006665777896280496
Confidence: 0.39062500000000006
Lift: 3.975682666214383
====================================
Rule: nan -> herb & pepper
Support: 0.015997866951073192
Confidence: 0.3234501347708895
Lift: 3.2919938411349285
====================================
Rule: spaghetti -> herb & pepper
Support: 0.006399146780429276
Confidence: 0.3934426229508197
Lift: 4.004359721511667
====================================
Rule: milk -> olive oil
Support: 0.004932675643247567
Confidence: 0.22424242424242427
Lift: 3.40494417862839
====================================
Rule: nan -> tomato sauce
Support: 0.005332622317024397
Confidence: 0.3773584905660377
Lift: 3.840659481324083
====================================
Rule: spaghetti -> shrimp
Support: 0.005999200106652446
Confidence: 0.5232558139534884
```