Project Title: Market Basket insights

Problem statement:

Unveiling Customer Behaviour through Association Analysis: Utilize market basket analysis on the provided dataset to uncover hidden patterns and associations between products, aiming to understand customer purchasing behaviour and identify potential cross-selling opportunities for the retail businesses.

Implementing market basket insights in python :

The method:

Here are the steps involved in using the apriori algorithm to implement market basket insights.

1.First, define the minimum support and confidence for the association rule.

2.Find out all the subsets in the transactions with higher support(sup) than the minimum support.

3.Find all the rules for these subsets with higher confidence than minimum confidence.

4.Sort these association rules in decreasing order.

5.Analyze the rules along with their confidence and support.

The Dataset:

The Apriori algorithm is frequently used by data scientists. We are required to import the necessary libraries. Python provides the apyori as an API that is required to be imported to run the Apriori Algorithm.

```
import pandas as pd
import numpy as np
from apyori import ap
```

There is no header in the dataset; hence, the first row contains the first transaction, so we have mentioned
header= None here.

```
import pandas as pd
import numpy as np
from apyori import apriori

st_df=pd.read_csv("store_data.csv",header=None)
print(st_df)
```

https://replit.com/@shivanshkausha/Businessanalysis

Once we have read the dataset completely, we are required to get the list of items in every transaction. So we are going to run two loops. One will be for the total number of transactions,

and the other will be for the total number of columns in every transaction. The list will work as a training set from where we can generate the list of Association Rules.

```
#converting dataframe into list of lists
l=[]
for i in range(1,7501):
l.append([str(st_df.values[i,j]) for j in range(0,20)])
```

So we are ready with the list of items in our training set, then we need to run the apriori algorithm, which will learn the list of association rules from the training set, i.e., list. So, the minimum support here will be 0.0045, which is taken here as support. Now let us see that we have kept 0.2 as the min confidence. The minimum lift value is taken as 3, and the minimum length is considered as 2 because we have to find an association among a minimum of two items.

```
#applying apriori algorithm
association_rules = apriori(l, min_support=0.0045, min_confidence=0.2, min_lift=3,
min_length=2)
association_results = list(association_rules)
```

After running the above line of code, we generated the list of association rules between the items. So to see these rules, the below line of code needs to be run.

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

Output:

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

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

Here we are going to display the Rule, Support, and lift ratio for every above association rule by using for loop.

```
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 position
    # of the third index of the inner list
    print("Confidence: " + str(item[2][0][2]))
    print("Lift: " + str(item[2][0][3]))
    print("----------------------------------------------------")
```

Output:

```
Rule: light cream -> chicken
Support: 0.004533333333333334
Confidence: 0.2905982905982906
Lift: 4.843304843304844
---------------------------------------------------
Rule: mushroom cream sauce -> escalope
Support: 0.005733333333333333
Confidence: 0.30069930069930073
Lift: 3.7903273197390845
---------------------------------------------------
Rule: pasta -> escalope
Support: 0.005866666666666667
Confidence: 0.37288135593220345
Lift: 4.700185158809287
---------------------------------------------------
Rule: herb & pepper -> ground beef
Support: 0.016
Confidence: 0.3234501347708895
Lift: 3.2915549671393096
---------------------------------------------------
Rule: tomato sauce -> ground beef
Support: 0.005333333333333333
Confidence: 0.37735849056603776
Lift: 3.840147461662528
---------------------------------------------------
Rule: whole wheat pasta -> olive oil
Support: 0.008
Confidence: 0.2714932126696833
Lift: 4.130221288078346
---------------------------------------------------
Rule: shrimp -> pasta
Support: 0.005066666666666666
Confidence: 0.3220338983050848
Lift: 4.514493901473151
---------------------------------------------------
Rule: nan -> light cream
```

Support: 0.004533333333333334
Confidence: 0.2905982905982906
Lift: 4.843304843304844

Rule: shrimp -> frozen vegetables
Support: 0.005333333333333333
Confidence: 0.23255813953488372
Lift: 3.260160834601174

Rule: spaghetti -> cooking oil
Support: 0.0048
Confidence: 0.5714285714285714
Lift: 3.281557646029315

Rule: mushroom cream sauce -> nan
Support: 0.005733333333333333
Confidence: 0.30069930069930073
Lift: 3.7903273197390845

Rule: nan -> pasta
Support: 0.005866666666666667
Confidence: 0.37288135593220345
Lift: 4.700185158809287

Rule: spaghetti -> frozen vegetables
Support: 0.008666666666666666
Confidence: 0.3110047846889952
Lift: 3.164906221394116

Rule: olive oil -> frozen vegetables
Support: 0.0048
Confidence: 0.20338983050847456
Lift: 3.094165778526489

Rule: shrimp -> frozen vegetables
Support: 0.0072
Confidence: 0.3068181818181818
Lift: 3.2183725365543547

Rule: spaghetti -> olive oil
Support: 0.005733333333333333
Confidence: 0.20574162679425836
Lift: 3.1299436124887174

Rule: spaghetti -> shrimp
Support: 0.006
Confidence: 0.21531100478468898
Lift: 3.0183785717479763

Rule: spaghetti -> frozen vegetables
Support: 0.006666666666666667
Confidence: 0.23923444976076555
Lift: 3.497579674864993

Rule: spaghetti -> grated cheese
Support: 0.005333333333333333
Confidence: 0.3225806451612903
Lift: 3.282706701098612

Rule: herb & pepper -> mineral water
Support: 0.006666666666666667
Confidence: 0.390625
Lift: 3.975152645861601

Rule: nan -> herb & pepper
Support: 0.016
Confidence: 0.3234501347708895
Lift: 3.2915549671393096

Rule: spaghetti -> herb & pepper
Support: 0.0064
Confidence: 0.3934426229508197
Lift: 4.003825878061259

Rule: olive oil -> milk
Support: 0.004933333333333333
Confidence: 0.22424242424242424
Lift: 3.411395906324912

Rule: nan -> tomato sauce
Support: 0.005333333333333333
Confidence: 0.37735849056603776
Lift: 3.840147461662528

Rule: spaghetti -> shrimp
Support: 0.006
Confidence: 0.5232558139534884
Lift: 3.004914704939635

Rule: spaghetti -> olive oil
Support: 0.0072
Confidence: 0.20300751879699247
Lift: 3.0883496774390333

Rule: soup -> olive oil
Support: 0.0052
Confidence: 0.2254335260115607
Lift: 3.4295161157945335

Rule: whole wheat pasta -> nan
Support: 0.008
Confidence: 0.2714932126696833
Lift: 4.130221288078346

Rule: nan -> shrimp
Support: 0.005066666666666666
Confidence: 0.3220338983050848
Lift: 4.514493901473151

Rule: spaghetti -> olive oil
Support: 0.005066666666666666
Confidence: 0.20105820105820105
Lift: 3.0586947422647217

Rule: nan -> shrimp
Support: 0.005333333333333333
Confidence: 0.23255813953488372
Lift: 3.260160834601174

Rule: spaghetti -> nan
Support: 0.0048
Confidence: 0.5714285714285714
Lift: 3.281557646029315

Rule: spaghetti -> nan
Support: 0.008666666666666666
Confidence: 0.3110047846889952
Lift: 3.164906221394116

Rule: spaghetti -> frozen vegetables
Support: 0.004533333333333334
Confidence: 0.28813559322033905

Lift: 3.0224013274860737

Rule: nan -> frozen vegetables
Support: 0.0048
Confidence: 0.20338983050847456
Lift: 3.094165778526489

Rule: nan -> shrimp
Support: 0.0072
Confidence: 0.3068181818181818
Lift: 3.2183725365543547

Rule: spaghetti -> nan
Support: 0.005733333333333333
Confidence: 0.20574162679425836
Lift: 3.1299436124887174

Rule: spaghetti -> nan
Support: 0.006
Confidence: 0.21531100478468898
Lift: 3.0183785717479763

Rule: spaghetti -> nan
Support: 0.006666666666666667
Confidence: 0.23923444976076555
Lift: 3.497579674864993

Rule: spaghetti -> nan
Support: 0.005333333333333333
Confidence: 0.3225806451612903
Lift: 3.282706701098612

Rule: nan -> herb & pepper
Support: 0.006666666666666667
Confidence: 0.390625
Lift: 3.975152645861601

Rule: spaghetti -> nan
Support: 0.0064
Confidence: 0.3934426229508197
Lift: 4.003825878061259

Rule: nan -> milk
Support: 0.004933333333333333

Confidence: 0.22424242424242424
Lift: 3.411395906324912

Rule: spaghetti -> nan
Support: 0.006
Confidence: 0.5232558139534884
Lift: 3.004914704939635

Rule: spaghetti -> nan
Support: 0.0072
Confidence: 0.20300751879699247
Lift: 3.0883496774390333

Rule: soup -> nan
Support: 0.0052
Confidence: 0.2254335260115607
Lift: 3.4295161157945335

Rule: spaghetti -> nan
Support: 0.005066666666666666
Confidence: 0.20105820105820105
Lift: 3.0586947422647217

Rule: spaghetti -> milk
Support: 0.004533333333333334
Confidence: 0.28813559322033905
Lift: 3.0224013274860737

Conclusion:

we discussed Market Basket insights and learned the steps to implement it from scratch using Python. We then implemented Market Basket Analysis using Apriori Algorithm. We also looked into the various uses and advantages of this algorithm.