MARKET BASKET INSIGHT

Abstract-Data mining refers to extracting knowledge from large amount of data and which is used for identifying the relation between one item to another. The association rule mining identifies relationship between a large set of data items. Finding of these relationships can help the retailers to develop a sales strategy by considering the items frequently purchased together by customers. This work acts as a wide area for the researchers to develop a better data mining algorithm. This research discussed the market basket analysis (MBA) by using apriori algorithm.

Keyword: Market basket analysis (MBA), Apriori Algorithm, Data Mining; Association Rule Mining

1. INTRODUCTION TO MARKET BASKET ANALYSIS

Market Basket Analysis (MBA) is one of the key methods used by vast retailers to uncover associations between items. It works by looking for groupings of items that occur together frequently in transactions. In another way, it allows retailers to identify relationships between the items that people buy. Association Rules are widely used to analyse retail basket or transaction data, and are intended to identify strong rules discovered in transaction data using measures of interest, based on the concept of strong rules. This research is a learning for identifying how the resulted concept, the processing of the rule, and the achieved rule. Therefore, this research gives a new learning from each of the step of the usage system until it forms the resulted system.[1]

2. Market Basket Analysis: An Overview

Association rule is a technique which is looking for a relationship among an item with other items. Association rule is usually used if and then such as if A then B and C, this shows if A then B and C. To determine the Association's rules, it needs to be specified the support and confidence to restrict whether the rule is interesting or not.[2][3]

Support: It is the number of transactions with both A and B divided by the total number of transactions. These rules are not beneficial for low support values.

$$Support = \frac{freq(A,B)}{N}$$

Confidence: It specifies how frequently the items A and B are bought together, for the no. of times A is bought.

$$Confidence = \frac{freq(A,B)}{freq(A)}$$

Lift: It specifies the strength of a rule over the randomness of A and B being bought together. It basically measures the strength of any association rule.[8]

$$Lift = \frac{Support}{Supp(A) \times Supp(B)}$$

Mining Association Rules:

Transaction ID	Items Bought
2000	A, B, C
1000	A, C
4000	A, D
5000	B, E, F

Table No 1: Sample Data set

1)For rule A, C

Support = support ({A &C})/Total TID = 2/4 = 50%

Confidence = support ({A &C})/support({A}) =2/3 = 66.6%

2)For Rule C, A:

Support = support ({C & A})/ Total TID =2/4= 50%

Confidence = support ({C & A})/support({C}) =2/2 = 100%.

3. INTRODUCTION OF DATA SET

In the implementation, the dataset used for Market Basket Analysis is the dataset that is publicly available on Kaggle. This dataset includes the list of transactions of a retail company over the period of one week. It contains a total of 9835 transaction records where each record consists of the list of items sold in one transaction.

4. MARKET BASKET ANALYSIS (MBA)

Apriori algorithm for discovery frequency item set. The Apriori algorithm analyses a data set to regulate which combinations of items occur together frequently. It is at the core of many algorithms for data mining problems. The best-known problem is discovery the association rules that hold in a basket item relation. Basic idea behind this algorithm is

- The item-set can only be a large item set if all its subsets are large item-sets.
- The sets of items that have lowest support can be considered.
- Association rules can be generated from frequent item sets.[6]

Step 1: Describe data set of Grocery item.

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	citrus fruit	semi-finished bread	margarine	ready soups	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	tropical fruit	yogurt	coffee	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	whole milk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	pip fruit	yogurt	cream cheese	meat spreads	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	other vegetables	whole milk	condensed milk	long life bakery product	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	whole milk	butter	yogurt	rice	abrasive cleaner	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	rolls/buns	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	other vegetables	UHT-milk	rolls/buns	bottled beer	liquor (appetizer)	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	pot plants	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	whole milk	cereals	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

10 rows × 32 columns

```
df.shape
```

```
(9835, 32)
```

Table No. 2: Data set

Step 2: Data Preparation prepares the data Once the data resources available are identified, they need to be selected, cleaned, built into the form desired, and formatted. Data cleaning and data transformation in preparation of data modeling needs to occur in this phase. Data exploration at a greater depth can be applied during this phase, and additional models utilized, again providing the opportunity to see patterns based on business understanding.[7],[8]

```
df_clean = df1.drop(['nan'], axis = 1)
df_clean
```

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics	baby food	bags	baking powder	bathroom cleaner	beef	...	turkey	vinegar	waffles	whipped/sour cream	whisky	white bread	whit wir
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9830	False	False	False	False	False	False	False	False	False	True	...	False	False	False	True	False	False	Fals
9831	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
9832	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
9833	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	Fals
9834	False	False	False	False	False	False	False	False	False	False	...	False	True	False	False	False	False	Fals

9835 rows × 169 columns

Table No.3: Data set After Data Preparation

Step 3: Find Top selling items and will visualize

	items	count
0	whole milk	2513
1	other vegetables	1903
2	rolls/buns	1808
3	soda	1713
4	yogurt	1372
5	bottled water	1086
6	root vegetables	1072
7	tropical fruit	1032
8	shopping bags	968
9	sausage	924
10	pastry	875
11	citrus fruit	814
12	bottled beer	789
13	newspapers	783
14	canned beer	764
15	pip fruit	744
16	fruit/vegetable juice	706
17	whipped/sour cream	705
18	brown bread	638
19	domestic eggs	623

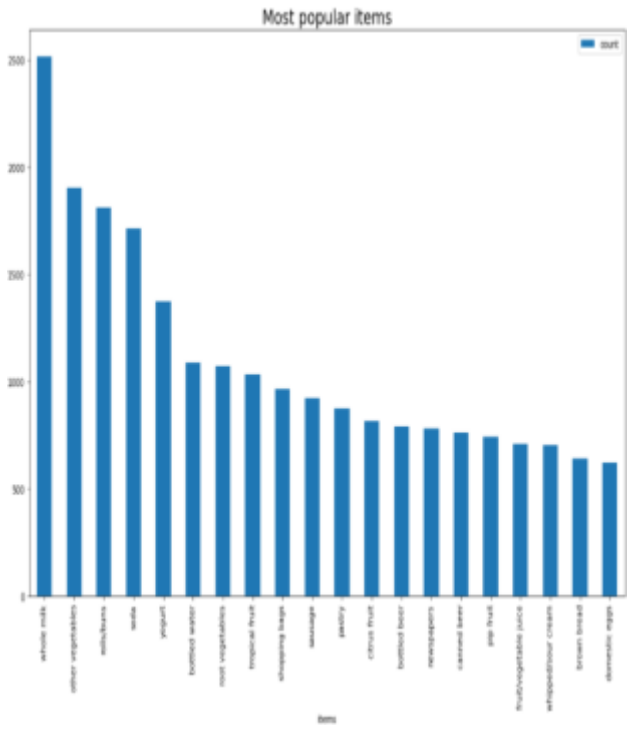

Table No:4 Top selling item.

Step 4: Find a frequent item set for two items in data set.

```
frequent_itemsets[ (frequent_itemsets['length'] == 2)]
```

	support	itemsets	length
13	0.160523	(other vegetables, whole milk)	2
14	0.121265	(rolls/buns, whole milk)	2
15	0.085714	(whole milk, soda)	2
16	0.120174	(whole milk, yogurt)	2
17	0.073501	(bottled water, whole milk)	2
...	...	...	...
85	0.026827	(pastry, sausage)	2
86	0.024209	(sausage, citrus fruit)	2
87	0.016576	(bottled beer, sausage)	2
88	0.020938	(pastry, citrus fruit)	2
89	0.013086	(bottled beer, citrus fruit)	2

77 rows × 3 columns

Table 5: Frequent Item set.

`

Step 5: Eliminate minimum support frequent itemset which are not found regularly .

```
def prune_dataset(olddf, len_transaction, tot_sales_percent):
    if 'tot_items' in olddf.columns:
        del(olddf['tot_items'])
    Item_count = olddf.sum().sort_values(ascending = False).reset_index()
    tot_items = sum(olddf.sum().sort_values(ascending = False))
    Item_count.rename(columns={Item_count.columns[0]:'Item_name',Item_count.columns[1]:'Item_count'}, inplace=True)
    Item_count['Item_percent'] = Item_count['Item_count']/tot_items
    Item_count['Tot_percent'] = Item_count.Item_percent.cumsum()
    selected_items = list(Item_count[Item_count.Tot_percent < tot_sales_percent].Item_name)
    olddf['tot_items'] = olddf[selected_items].sum(axis = 1)
    olddf = olddf[olddf.tot_items >= len_transaction]
    del(olddf['tot_items'])
    return olddf[selected_items], Item_count[Item_count.Tot_percent < tot_sales_percent]
```

```
output_df, item_counts = prune_dataset(df_clean,
2,0.4)
print(output_df.shape)
print(list(output_df.columns))
output_df
```

```
(4585, 13)
['whole milk', 'other vegetables', 'rolls/buns', 'soda', 'yogurt', 'bottled water', 'root vegetables', 'tropical fruit', 'shopp
ing bags', 'sausage', 'pastry', 'citrus fruit', 'bottled beer']
```

Table no 6: Pruning frequent itemset

Step no:6 Implementing Apriori Algorithm.

```
rules_mlxtend[ (rules_mlxtend['antecedent_len'] >= 2) &
       (rules_mlxtend['confidence'] >= 0.3) &
       (rules_mlxtend['lift'] >= 1) ]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len
48	(yogurt, other vegetables)	(whole milk)	0.093130	0.442748	0.047764	0.512881	1.158403	0.006531	1.143974	2
49	(yogurt, whole milk)	(other vegetables)	0.120174	0.352454	0.047764	0.397459	1.127692	0.005409	1.074693	2
54	(other vegetables, whole milk)	(root vegetables)	0.160523	0.207852	0.049727	0.309783	1.490402	0.016362	1.147679	2
55	(other vegetables, root vegetables)	(whole milk)	0.101636	0.442748	0.049727	0.489270	1.105076	0.004728	1.091090	2
56	(whole milk, root vegetables)	(other vegetables)	0.104907	0.352454	0.049727	0.474012	1.344893	0.012752	1.231106	2
...	...	...	...	...	...	...	...	...	...	...
547	(whole milk, tropical fruit, root vegetables)	(yogurt)	0.025736	0.256052	0.012214	0.474576	1.853435	0.005624	1.415900	3
558	(yogurt, other vegetables, root vegetables)	(tropical fruit)	0.027699	0.199346	0.010687	0.385827	1.935466	0.005165	1.303629	3
559	(yogurt, other vegetables, tropical fruit)	(root vegetables)	0.026390	0.207852	0.010687	0.404959	1.948306	0.005202	1.331249	3
560	(yogurt, root vegetables, tropical fruit)	(other vegetables)	0.017448	0.352454	0.010687	0.612500	1.737817	0.004537	1.671087	3
561	(other vegetables, root vegetables, tropical f...	(yogurt)	0.026390	0.256052	0.010687	0.404959	1.581546	0.003930	1.250245	3

Table no 7: Implementing Apriori Algorithm-using Associations Rules

5. CONCLUSION:

In this paper researcher mainly understood and focused on Apriori algorithm how it was predicting the frequent itemset. Apriori produce association rules for better prediction of customer behaviors penetrated items show a correlation between the data and information of support and the confidence that can be analyzed. This information will give additional consideration for the user to make further decision making & also retailer and super market will be implemented in their business for maintaining their stock according with association rules and it will give phenomenal benefits to particular business.