VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

PROJECT WORK-4 REPORT
on

“Market Basket Analysis”

Submitted by

Abhishek R (1BM19CS400)
Arbaz Ahmed(1BM19CS401)
Harsha R(1BM18CS034)
Kaushal Jalan (1BM18CS136)

Under the Guidance of
Dr. Kavitha Sooda
Assistant Professor, BMSCE

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Mar-2021 to Jun-2021

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled "Market Basket Analysis" carried out by Abhishek R (1BM19CS400),Arbaz Ahmed (1BM19CS401), Harsha R (1BM18CS034),AND Kaushal Jalan (1BM18CS146) who are bonafide students of B. M. S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2020-2021. The project report has been approved as it satisfies the academic requirements in respect of Project Work-4 (20CS6PWPW4) work prescribed for the said degree.

Signature of the Guide	Signature of the HOD
Dr.Kavitha Sooda	Dr. Umadevi V
Associate Professor	Associate Prof. & Head, Dept. of CSE
BMSCE, Bengaluru	BMSCE, Bengaluru

External Viva

Name of the Examiner	Signature with date
1.______________________________	________________________
2. ______________________________	_________________________

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECALARATION

We, Abhishek R (1BM19CS400),Arbaz Ahmed (1BM19CS401), Harsha R (1BM18CS038), Kaushal Jalan (1BM18CS146), students of 6th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, here by declare that, this Project Work-4 entitled "Market Basket Analysis" has been carried out by us under the guidance of Dr.Kavitha Sooda, Associate Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester Mar-2021-Jun-2021

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

Abhishek R (1BM18CS400)

Arbaz Ahmed (1BM19CS401)

Harsha R (1BM18CS034)

Kaushal Jalan (1BM19CS146)

INTRODUCTION

Market Basket Analysis(MBA) also known as association rule learning or affinity analysis, is a data mining technique that can be used in various fields, such as marketing, bioinformatics, education field, nuclear science etc. The main aim of MBA in marketing is to provide the information to the retailer to understand the purchase behavior of the buyer, which can help the retailer in correct decision making. There are various algorithms are available for performing MBA. The existing algorithms work on static data and they do not capture changes in data with time. But proposed algorithm not only mine static data but also provides a new way to take into account changes happening in data. This paper discusses the data mining technique i.e. association rule mining and provide a new algorithm which may helpful to examine the customer behaviour and assists in increasing the sales.

Today, the large amount of data is being maintained in the databases in various fields like retail markets, banking sector, medical field etc. But it is not necessary that the whole information is useful for the user. That is why, it is very important to extract the useful information from large amount of data. This process of extracting useful data is known as data mining or A Knowledge Discovery and Data (KDD) process. The overall process of finding and interpreting patterns from data involves many steps such as selection, preprocessing, transformation, data mining and interpretation. Data mining helps in the business for marketing. The work of using market basket analysis in management research has been performed by Aguinis et al. Market basket analysis is also known as association rule mining. It helps the marketing analyst to understand the behavior of customers e.g. which products are being bought together. There are various techniques and algorithms that are available to perform data mining.

Problem Definition and Algorithm

Task Definition

When you go to the supermarket, usually the first thing you do is grab a shopping cart. As you move up and down the aisles, you will pick up certain items and place them in your shopping cart. Most of these items may correspond to a shopping list that was prepared ahead of time, but other items may have been selected spontaneously. Let's presume that when you check out at the cashier, the contents of your (and every other shopper's) cart are logged, because the supermarket wants to see if there are any patterns in selection that occur from one shopper to another.

Market basket analysis is a process that looks for relationships of objects that "go together" within the business context. In reality, market basket analysis goes beyond the supermarket scenario from which its name is derived. Market basket analysis is the analysis of any collection of items to identify affinities that can be exploited in some manner.

Algorithm Definition

The Apriori algorithm uses frequent item sets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a breadth-first search and Hash Tree to calculate the item set associations efficiently. It is the iterative process for finding the frequent item sets from the large dataset.

This algorithm was given by the R. Agrawal and Srikant in the year 1994. It is mainly used for market basket analysis and helps to find those products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

Below are the steps for the apriori algorithm:

- Determine the support of itemsets in the transactional database, and select the minimum support and confidence.
- Take all supports in the transaction with higher support value than the minimum or selected support value.
- Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.
- Sort the rules as the decreasing order of lift.

Experimental Evaluation

To carry out an MBA you'll first need a data set of transactions. Each transaction represents a group of items or products that have been bought together and often referred to as an "itemset". For example, one itemset might be: {pencil, paper, staples, rubber} in which case all of these items have been bought in a single transaction.

In an MBA, the transactions are analysed to identify rules of association. For example, one rule could be: {pencil, paper} => {rubber}. This means that if a customer has a transaction that contains a pencil and paper, then they are likely to be interested in also buying a rubber.
Before acting on a rule, a retailer needs to know whether there is sufficient evidence to suggest that it will result in a beneficial outcome. We therefore measure the strength of a rule by calculating the following three metrics (note other metrics are available, but these are the three most commonly used):

Support: the percentage of transactions that contain all of the items in an itemset (e.g., pencil, paper and rubber). The higher the support the more frequently the itemset occurs. Rules with a high support are preferred since they are likely to be applicable to a large number of future transactions.

Confidence: the probability that a transaction that contains the items on the left hand side of the rule (in our example, pencil and paper) also contains the item on the right hand side (a rubber). The higher the confidence, the greater the likelihood that the item on the right hand side will be purchased or, in other words, the greater the return rate you can expect for a given rule.

Lift: the probability of all of the items in a rule occurring together (otherwise known as the support) divided by the product of the probabilities of the items on the left and right hand side occurring as if there was no association between them. For example, if pencil, paper and rubber occurred together in 2.5% of all transactions, pencil and paper in 10% of transactions and rubber in 8% of transactions, then the lift would be: 0.025/(0.1*0.08) = 3.125. A lift of more than 1 suggests that the presence of pencil and paper increases the probability that a rubber will also occur in the transaction. Overall, lift summarises the strength of association between the products on the left and right hand side of the rule; the larger the lift the greater the link between the two products.

To perform a Market Basket Analysis and identify potential rules, a data mining algorithm called the 'Apriori algorithm' is commonly used, which works in two steps:

1. Systematically identify itemsets that occur frequently in the data set with a support greater than a pre-specified threshold.
2. Calculate the confidence of all possible rules given the frequent itemsets and keep only those with a confidence greater than a pre-specified threshold.

The thresholds at which to set the support and confidence are user-specified and are likely to vary between transaction data sets. R does have default values, but we recommend that you experiment with these to see how they affect the number of rules returned (more on this below). Finally, although the Apriori algorithm does not use lift to establish rules, you'll see in the following that we use lift when exploring the rules that the algorithm returns.

Methodology

Let us consider the first row of the table.

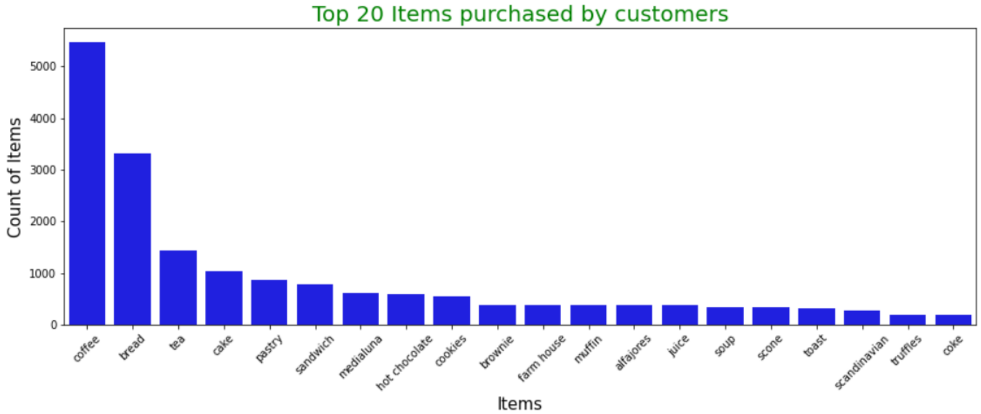
It can be read as Toast → Coffee {antecedents → consequents}

It stated that if Toast then Coffee which means when Toast is ordered people also ordered Coffee and to support this statement we have confidence and lift which are 0.70 and 1.47 respectively which are very good.

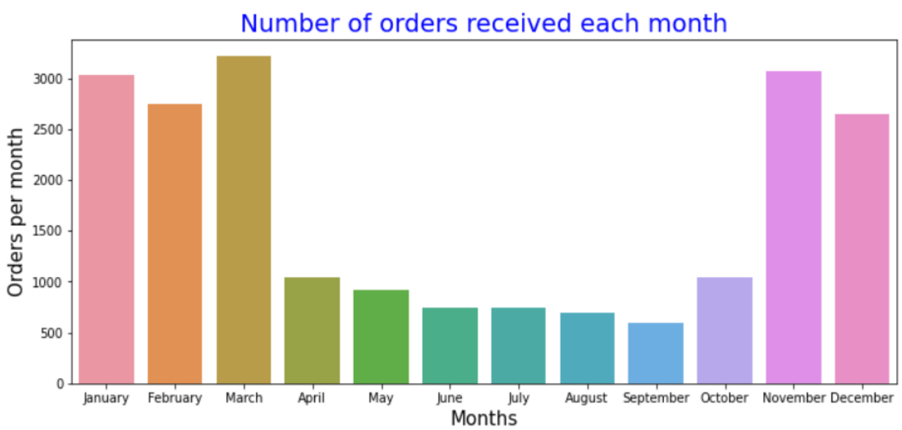
- antecedents: It is an item (here toast) who support the other item (coffee).
- consequents: It is an item (here coffee) who is supported by the an item (toast).
- antecedent support: Support of an antecedent (toast). It states that 3.36% of transactions contain toast.
- consequent support: Support of consequent (coffee). It states that 47.84% of transactions contain coffee.
- support: Support of an both antecedent and consequent. It states that 2.37% of transactions contain both toast and coffee.
- confidence: A confidence of 0.7 would mean that in 70% of the cases where toast were purchased, the purchase also included coffee.
- lift: Lift of greater than 1 means products A and B are more likely to be bought together. Here lift of 1.47 means the likelihood of a customer buying both toast and coffee together is 1.47 times more than the chance of purchasing coffee alone.

Results

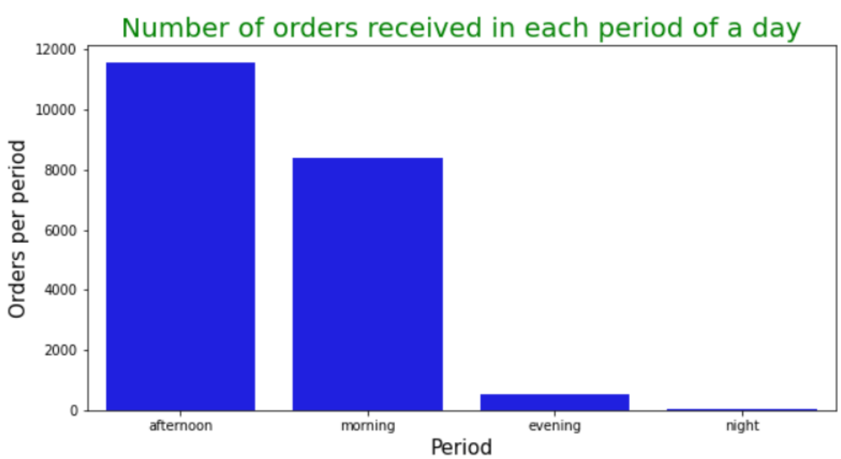
1.Top items purchased



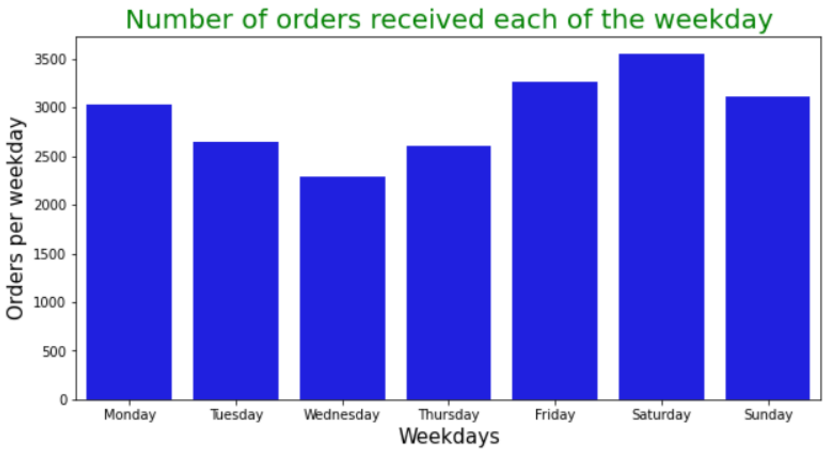
2.Monthly Transcations



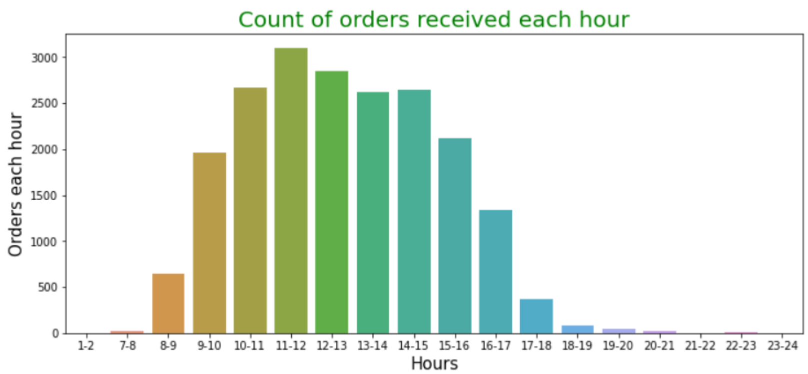
3.Number of orders per period



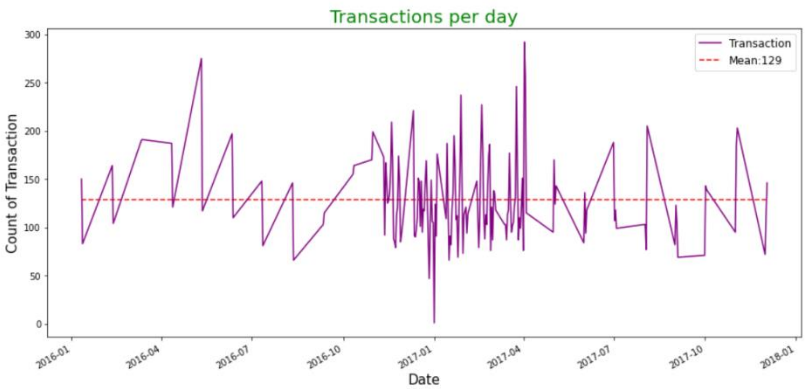
4.Number of orders per weekday



5.Count of orders received each hour.

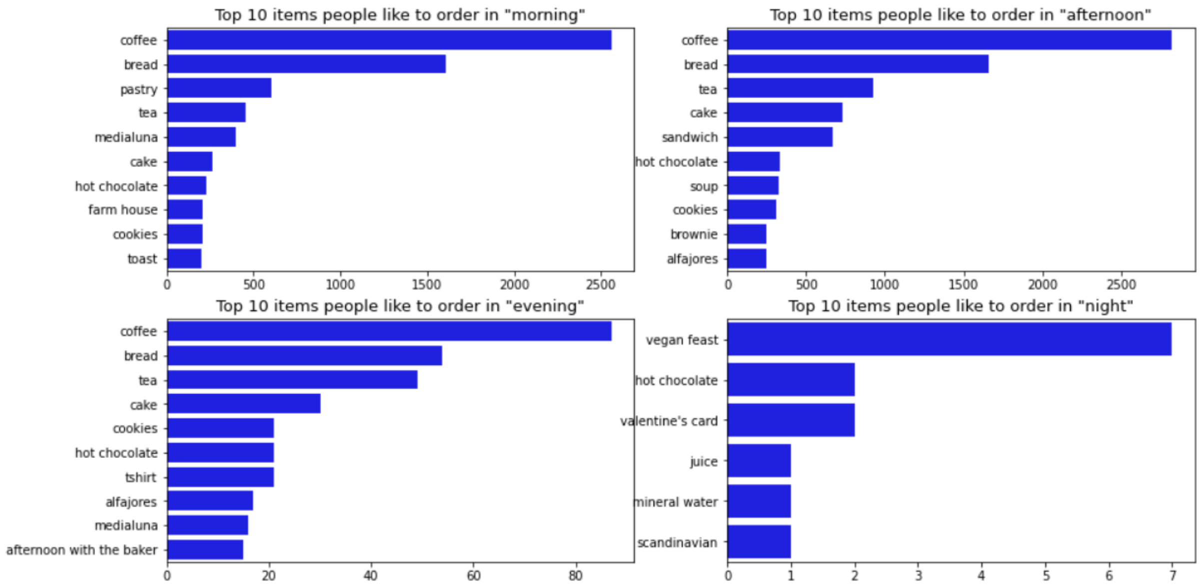


6.Count of transactions per day



<u>7.Items people like to order in morning,afternoon,evening,night</u>

]



Discussion

One way in which you can relate these concepts is to think in the following way:

- Support is similar to Recall metric, a rule with high support means it has high presence on the dataset.
- Confidence is similar to Precision metric, a rule with high confidence means it has high precision whenever the rule appears.

  While mining association rules,you prefer the ones with high confidence (precision) over ones with high support (recall). That is why usually the minSupport may be on the lower side, even lower than 50%, while minConfidence is usually set higher, above 50% for example.

On the basis of high confidence (considered greater than or equal to 0.55), we have the following rules:

- toast → coffee
- spanish brunch → coffee
- medialuna → coffee
- pastry → coffee

  Also note that these rules have lift > 1 which means those pairs have positive correlation between them.

  It is been observed that (coffee, tea) → (cake) has a highest lift of 1.94 which indicates that they have high correlation between them. Here lift of 1.94 means the likelihood of a customer buying all coffee, tea and cake together is 1.94 times more than the chance of purchasing cake alone.

Related work

Topic: Market Basket Analysis

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import association_rules, apriori
```

```
data = pd.read_csv('bread_basket.csv')
data.head()
```

	Transaction	Item	date_time	period_day	weekday_weekend
0	1	Bread	30-10-2016 09:58	morning	weekend
1	2	Scandinavian	30-10-2016 10:05	morning	weekend
2	2	Scandinavian	30-10-2016 10:05	morning	weekend
3	3	Hot chocolate	30-10-2016 10:07	morning	weekend
4	3	Jam	30-10-2016 10:07	morning	weekend

```
num_transactions = data['Transaction'].unique().shape[0]
num_transactions
```

```
9465
```

```
unique_items = data['Item'].unique()
num_items = unique_items.shape[0]
print(num_items)
unique_items.reshape(num_items,1)
```

```
94
array([['bread'],
       ['scandinavian'],
       ['hot chocolate'],
       ['jam'],
       ['cookies'],
       ['muffin'],
       ['coffee'],
       ['pastry'],
       ['medialuna'],
       ['tea'],
       ['tartine'],
       ['basket'],
       ['mineral water'],
       ['farm house'],
       ['fudge'],
       ['juice'],
       ["ella's kitchen pouches"],
       ['victorian sponge'],
       ['frittata'],
       ['hearty & seasonal'],
       ['soup'],
```

```
       ['pick and mix bowls'],
       ['smoothies'],
       ['cake'],
       ['mighty protein'],
       ['chicken sand'],
       ['coke'],
       ['my-5 fruit shoot'],
       ['focaccia'],
       ['sandwich'],
       ['alfajores'],
       ['eggs'],
       ['brownie'],
       ['dulce de leche'],
       ['honey'],
       ['the bart'],
       ['granola'],
       ['fairy doors'],
       ['empanadas'],
       ['keeping it local'],
       ['art tray'],
       ['bowl nic pitt'],
       ['bread pudding'],
       ['adjustment'],
       ['truffles'],
       ['chimichurri oil'],
       ['bacon'],
       ['spread'],
       ['kids biscuit'],
       ['siblings'],
       ['caramel bites'],
       ['jammie dodgers'],
       ['tiffin'],
       ['olum & polenta'],
       ['polenta'],
       ['the nomad'],
       ['hack the stack'],
       ['bakewell'],
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20507 entries, 0 to 20506
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Transaction      20507 non-null  int64
 1   Item             20507 non-null  object
 2   date_time        20507 non-null  object
 3   period_day       20507 non-null  object
 4   weekday_weekend  20507 non-null  object
dtypes: int64(1), object(4)
memory usage: 801.2+ KB
```

```
data['date_time'] = pd.to_datetime(data['date_time'])
data.head()
```

Transaction Item date_time period_day weekday_weekend

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20507 entries, 0 to 20506
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Transaction      20507 non-null  int64
 1   Item             20507 non-null  object
 2   date_time        20507 non-null  datetime64[ns]
 3   period_day       20507 non-null  object
 4   weekday_weekend  20507 non-null  object
dtypes: datetime64[ns](1), int64(1), object(3)
memory usage: 801.2+ KB
```

Data Preprocessing

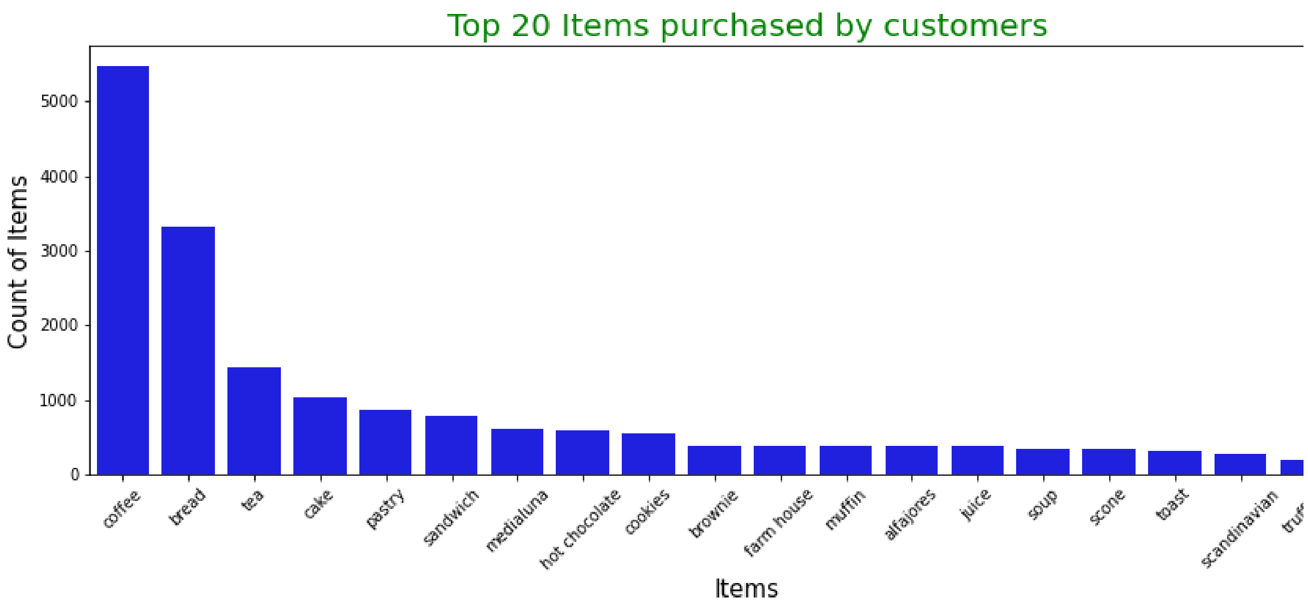
```
data['date'] = data['date_time'].dt.date
data['date'] = pd.to_datetime(data['date'], format = '%Y-%m-%d')

# Extracting time
data['time'] = data['date_time'].dt.time

# Extracting month and replacing it with text
data['month'] = data['date_time'].dt.month
data['month'] = data['month'].replace((1,2,3,4,5,6,7,8,9,10,11,12),
                                      ('January','February','March','April','May','June','July',
                                      'September','October','November','December'))

# Extracting hour
data['hour'] = data['date_time'].dt.hour
# Replacing hours with text
hour_in_num = (1,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23)
hour_in_obj = ('1-2','7-8','8-9','9-10','10-11','11-12','12-13','13-14','14-15',
               '15-16','16-17','17-18','18-19','19-20','20-21','21-22','22-23','23-24')
data['hour'] = data['hour'].replace(hour_in_num, hour_in_obj)

# Extracting weekday and replacing it with text
data['weekday'] = data['date_time'].dt.weekday
data['weekday'] = data['weekday'].replace((0,1,2,3,4,5,6),
                                      ('Monday','Tuesday','Wednesday','Thursday','Friday','Satur

# dropping date_time column
data.drop('date_time', axis = 1, inplace = True)

data['Item'] = data['Item'].str.strip()
data['Item'] = data['Item'].str.lower()

data.head()
```

	Transaction	Item	period_day	weekday_weekend	date	time	month	hour
0	1	bread	morning	weekend	2016-10-30	09:58:00	October	9-10

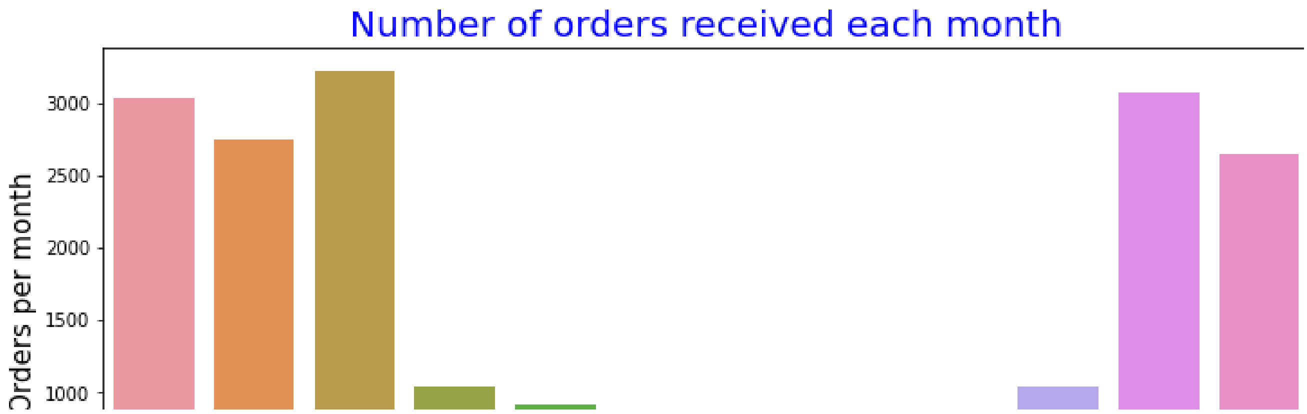
Data Visualizatoin

3	3	hot chocolate	morning	weekend	2016-10-30	10:07:00	October	10-11

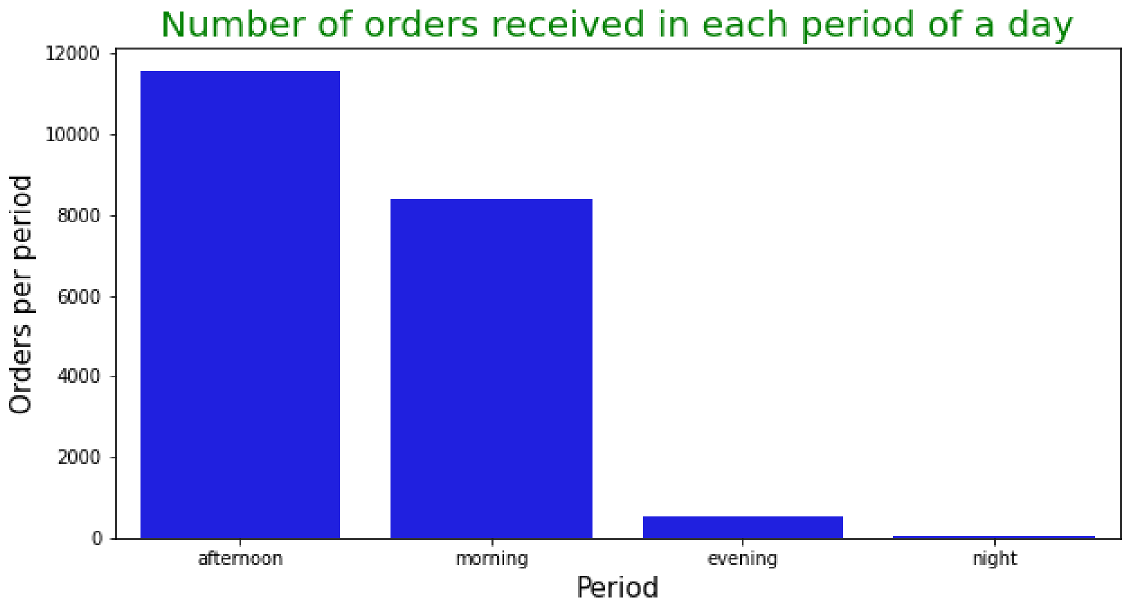
```
# top purchased items

plt.figure(figsize=(15,5))
sns.barplot(x = data.Item.value_counts().head(20).index, y = data.Item.value_counts().head(20).value
plt.xlabel('Items', size = 15)
plt.xticks(rotation=45)
plt.ylabel('Count of Items', size = 15)
plt.title('Top 20 Items purchased by customers', color = 'green', size = 20)
plt.show()
```




```
# monthly transaction

monthTran = data.groupby('month')['Transaction'].count().reset_index()
monthTran.loc[:,"monthorder"] = [4,8,12,2,1,7,6,3,5,11,10,9]
monthTran.sort_values("monthorder",inplace=True)


plt.figure(figsize=(12,5))
sns.barplot(data = monthTran, x = "month", y = "Transaction")
plt.xlabel('Months', size = 15)
plt.ylabel('Orders per month', size = 15)
plt.title('Number of orders received each month', color = 'blue', size = 20)
plt.show()
```

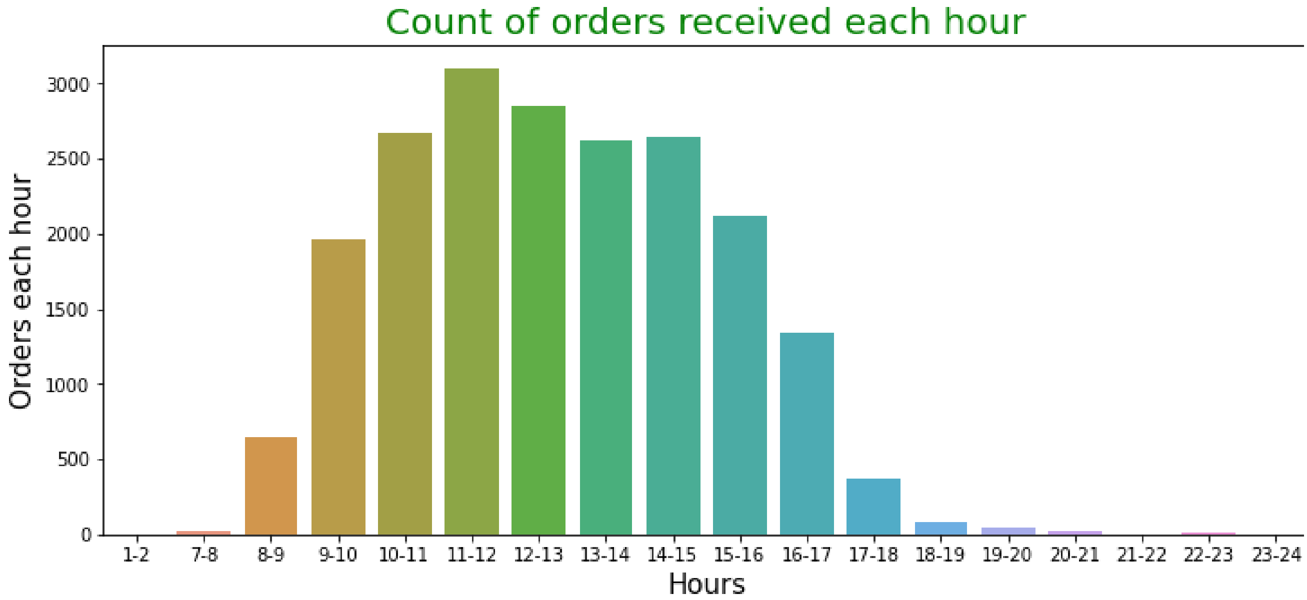



```
plt.figure(figsize=(10,5))
sns.barplot(x = data.period_day.value_counts().index, y = data.period_day.value_counts().values, col
plt.xlabel('Period', size = 15)
plt.ylabel('Orders per period', size = 15)
plt.title('Number of orders received in each period of a day', color = 'green', size = 20)
plt.show()
```

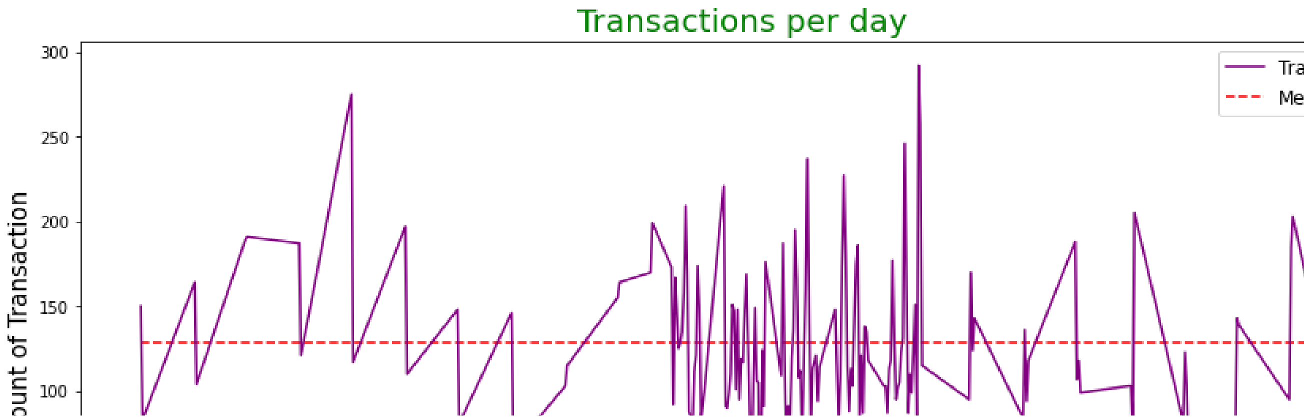



```
weekTran = data.groupby('weekday')['Transaction'].count().reset_index()
weekTran.loc[:,"weekorder"] = [5,1,6,7,4,2,3]
weekTran.sort_values("weekorder",inplace=True)


plt.figure(figsize=(10,5))
sns.barplot(data = weekTran, x = "weekday", y = "Transaction", color='blue')
plt.xlabel('Weekdays', size = 15)
plt.ylabel('Orders per weekday', size = 15)
plt.title('Number of orders received each of the weekday', color = 'green', size = 20)
plt.show()
```

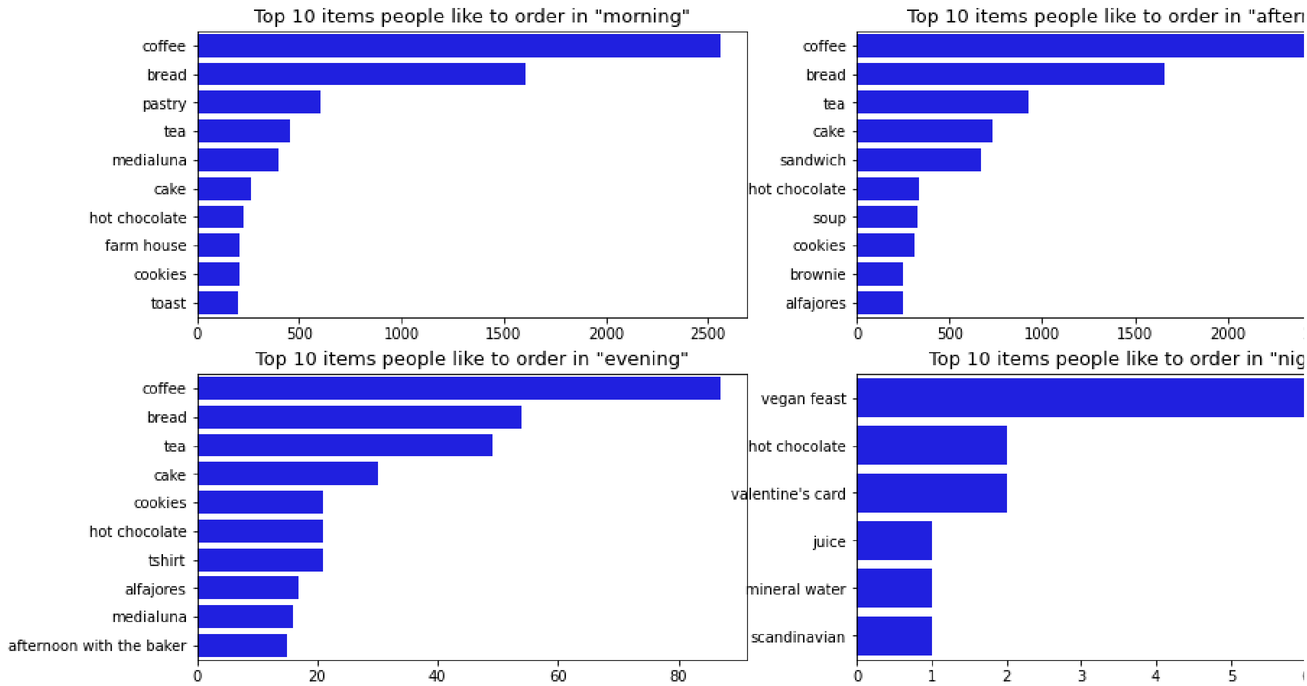

```
hourTran = data.groupby('hour')['Transaction'].count().reset_index()
hourTran.loc[:,"hourorder"] = [1,10,11,12,13,14,15,16,17,18,19,20,21,22,23,7,8,9]
hourTran.sort_values("hourorder",inplace=True)

plt.figure(figsize=(12,5))
sns.barplot(data = hourTran, x = "hour", y = "Transaction")
plt.xlabel('Hours', size = 15)
plt.ylabel('Orders each hour', size = 15)
plt.title('Count of orders received each hour', color = 'green', size = 20)
plt.show()
```




```
data.groupby('date')['Transaction'].count().plot(kind="line",figsize=(15,7),color='purple')
plt.xlabel('Date', size = 15)
plt.ylabel('Count of Transaction', size = 15)
plt.hlines(y = 129, color='red', xmin=data['date'].min(), xmax=data['date'].max(),
           linestyles='dashed', label='Mean:129')
plt.title('Transactions per day',  color = 'green', size = 20)
plt.legend(fontsize='large')
plt.show()
```




```
df = data.groupby(['period_day','Item'])['Transaction'].count().reset_index().sort_values(['period_d
day = ['morning','afternoon','evening','night']

plt.figure(figsize=(15,8))
for i,j in enumerate(day):
    plt.subplot(2,2,i+1)
    df1 = df[df.period_day==j].head(10)
    sns.barplot(data=df1, y=df1.Item, x=df1.Transaction, color='blue')
    plt.xlabel('')
    plt.ylabel('')
    plt.title('Top 10 items people like to order in "{}"'.format(j), size=13)

plt.show()
```



Associastion Rules mining using Apriori algorithm

```
df = data.groupby(['Transaction','Item'])['Item'].count().reset_index(name='Count')
df
```

	Transaction	Item	Count
0	1	bread	1
1	2	scandinavian	2
2	3	cookies	1
3	3	hot chocolate	1
4	3	jam	1
...	...	...	...
18882	9682	tacos/fajita	1
18883	9682	tea	1
18884	9683	coffee	1
18885	9683	pastry	1
18886	9684	smoothies	1

18887 rows × 3 columns

```
my_basket = df.pivot_table(index='Transaction', columns='Item', values='Count', aggfunc='sum').filln
my_basket.head()
```

Item	adjustment	afternoon with the baker	alfajores	argentina night	art tray	bacon	baguette	bakewell	b popc
Transaction									
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 94 columns

```
def encode(x):
    if x<=0:
        return 0
    if x>=1:
        return 1

# applying the function to the dataset

my_basket_sets = my_basket.applymap(encode)
my_basket_sets.head()
```

Item	adjustment	afternoon with the baker	alfajores	argentina night	art tray	bacon	baguette	bakewell	b popc
Transaction									
1	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	

```
frequent_itemsets = apriori(my_basket_sets, min_support = 0.01, use_colnames = True)
frequent_itemsets
```

	support	itemsets
0	0.036344	(alfajores)
1	0.016059	(baguette)
2	0.327205	(bread)
3	0.040042	(brownie)
4	0.103856	(cake)
...	...	...
56	0.023666	(toast, coffee)
57	0.014369	(sandwich, tea)
58	0.010037	(bread, cake, coffee)
59	0.011199	(bread, pastry, coffee)
60	0.010037	(cake, tea, coffee)

61 rows × 2 columns

```
rules = association_rules(frequent_itemsets, metric = "lift", min_threshold = 1)
rules.sort_values('confidence', ascending = False, inplace = True)
rules.head()
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
30	(toast)	(coffee)	0.033597	0.478394	0.023666	0.704403	1.472431	0.007593
29	(spanish brunch)	(coffee)	0.018172	0.478394	0.010882	0.598837	1.251766	0.002189
18	(medialuna)	(coffee)	0.061807	0.478394	0.035182	0.569231	1.189878	0.005614
22	(pastry)	(coffee)	0.086107	0.478394	0.047544	0.552147	1.154168	0.006351

```
rules.sort_values('confidence', ascending=False)
rules.head()
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
30	(toast)	(coffee)	0.033597	0.478394	0.023666	0.704403	1.472431	0.007593
29	(spanish brunch)	(coffee)	0.018172	0.478394	0.010882	0.598837	1.251766	0.002189
18	(medialuna)	(coffee)	0.061807	0.478394	0.035182	0.569231	1.189878	0.005614
22	(pastry)	(coffee)	0.086107	0.478394	0.047544	0.552147	1.154168	0.006351

```
rules.sort_values('lift',ascending=False)
rules.head()
```


	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
30	(toast)	(coffee)	0.033597	0.478394	0.023666	0.704403	1.472431	0.007593
29	(spanish brunch)	(coffee)	0.018172	0.478394	0.010882	0.598837	1.251766	0.002189
18	(medialuna)	(coffee)	0.061807	0.478394	0.035182	0.569231	1.189878	0.005614
22	(pastry)	(coffee)	0.086107	0.478394	0.047544	0.552147	1.154168	0.006351

Conclusion

At present many data mining algorithms have been developed and applied on variety of practical problems. However periodic mining is a new approach in data mining which has gained its significance these days. This field is evolving due to needs in different applications and limitations of data mining. This would enhance the power of existing data mining techniques. Finding out the patterns due to changes in data is in itself an interesting area to be explored. It may helpful in

- Find out interesting patterns from large amount of data.
- Automatically track the changes in facts from previous data; due to this feature it may be helpful in fraud detection.
- Predicting future association rules as well as gives us right methodology to find out outliers.

Authors suggested that, some areas are still there which need to be focused on. Firstly, results have influenced greatly by the manual threshold values for score, so it is needed to automate the threshold values for better recognition of outliers. Secondly, this approach is specifically targeted at Market Basket Data, it may perhaps be extended to other areas.

Bilbiography

[1] Market Basket Analysis: Identify the changing trends of market data using association rule mining, Manpreet Kaura , Shivani Kang