Market basket analysis for a bakery

Our project aims to enhance sales, marketing and customer satisfaction by analyzing ideas to improve product offerings.

Omar Alamoudi
00967-701590735
amryalamoudui99@gmail.com

Tools used in this project:

- SQL server 2019
- Microsoft Excel
- Python Programming language
- Microsoft Power Bi
- Microsoft PowerPoint

Data used in this project:

- https://www.kaggle.com/datasets/akashdeepkuila/bakery

context:

In today's digital age, even small businesses are operating online due to the huge opportunities available. As more people have access to the Internet and shop online, major retailers are looking for ways to increase their profits. To understand their customers' buying behaviors and patterns, retailers use market basket analysis, which examines groups of items to discover associations between products that are often purchased together.

Our goal for this project is Market basket analysis to improve sales and marketing strategies, enhance customer satisfaction, and improve product offerings based on the insights we gain from this analysis.

The questions we need answered:

1) What items are frequently purchased together?

2) How often do customers buy certain items together?

3) Can we upsell or cross-sell certain items based on customers' buying patterns?

By answering these questions, market basket analysis can help companies identify opportunities to improve sales and marketing strategies, improve product offerings, and enhance customer satisfaction.

In this case, the “items” variable is the key variable for market basket analysis.

Obtain data (O)

We have used the OSEMN framework in our project

We obtained our dataset for analysis from the Kaggle website, which contains transactional data for "The Bread Basket", a bakery located in Edinburgh. The dataset provides details of customers who ordered various items from the bakery online during the time period from 26-01-11 to 27-12-03. The dataset contains 20,507 entries, with over 9,000 transactions, and 4 columns. Columns include transaction number, items, date and time, and day part.

Obtain data (O)

Variables contained in the data set:

1. Transaction No: A unique identifier for each transaction

2. Items: Purchased items

3. DateTime: The date and time stamp of the transaction

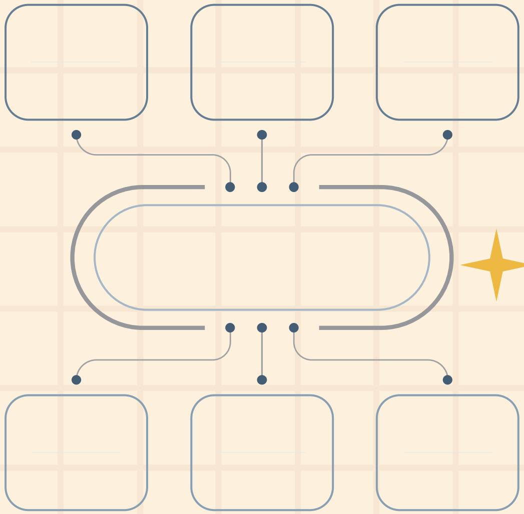
4. day part: the part of the day in which the transaction takes place (morning, afternoon, evening, night)

5. DayType: Classifies whether the transaction took place on a weekend or a weekday

Obtain data (O)

Overall, we believe this dataset is a good fit for our market basket analysis project, as it provides transactional data for a bakery that will allow us to identify patterns of customer buying behavior and make recommendations for product placement and promotions. Our goal for this project is to improve sales and marketing strategies, enhance customer satisfaction, and improve product offerings based on the insights we gain from this analysis.

We loaded the data into SQL Server to perform data cleaning steps and to troubleshoot.



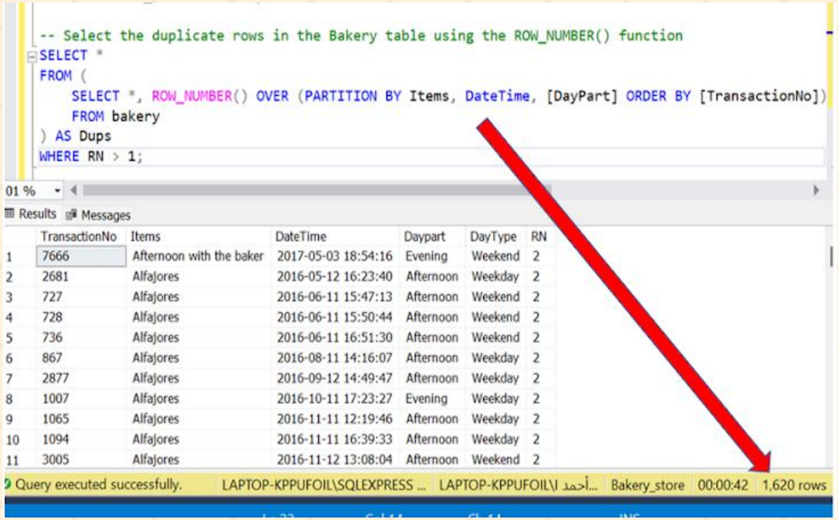
Scrub data (S)

The goal of data cleaning in the OSEMN framework is to ensure that the data set is accurate, complete, consistent, and ready for analysis.

By performing data cleaning steps, we can improve the quality of our analysis results:

1) Remove Duplicates: Identify any duplicate records in the data set and remove them.A more efficient way is to use ROW_NUMBER() to remove duplicate records.

We notice that there are 1620 duplicate rows in the table There are 20,507 rows in the data set, 1,620 rows are duplicates, so there are only 18,887 unique rows.



```sql
-- Select the duplicate rows in the Bakery table using the ROW_NUMBER() function
SELECT *
FROM (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY Items, DateTime, [DayPart] ORDER BY [TransactionNo])
    FROM bakery
) AS Dups
WHERE RN > 1;
```

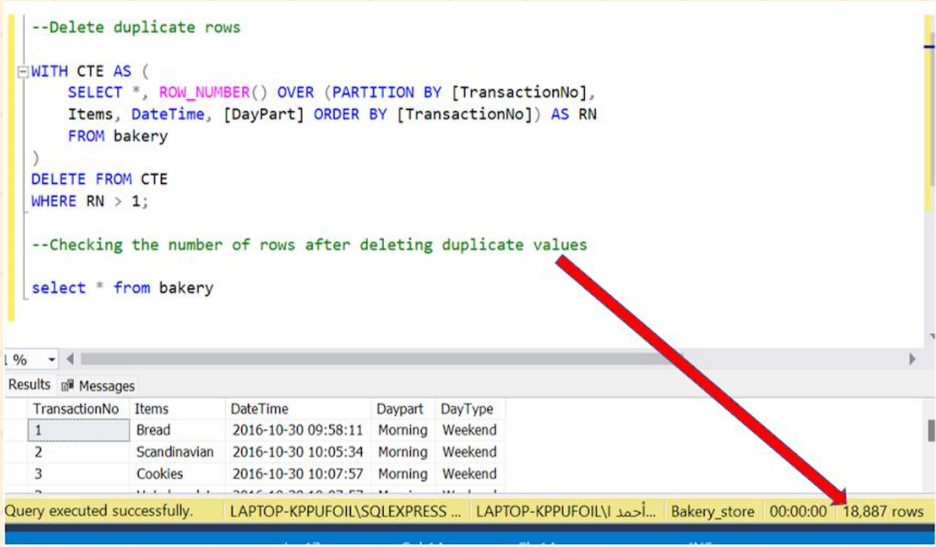
	TransactionNo	Items	DateTime	Daypart	DayType	RN
1	7666	Afternoon with the baker	2017-05-03 18:54:16	Evening	Weekend	2
2	2681	Alfajores	2016-05-12 16:23:40	Afternoon	Weekday	2
3	727	Alfajores	2016-06-11 15:47:13	Afternoon	Weekend	2
4	728	Alfajores	2016-06-11 15:50:44	Afternoon	Weekend	2
5	736	Alfajores	2016-06-11 16:51:30	Afternoon	Weekend	2
6	867	Alfajores	2016-08-11 14:16:07	Afternoon	Weekday	2
7	2877	Alfajores	2016-09-12 14:49:47	Afternoon	Weekday	2
8	1007	Alfajores	2016-10-11 17:23:27	Evening	Weekday	2
9	1065	Alfajores	2016-11-11 12:19:46	Afternoon	Weekday	2
10	1094	Alfajores	2016-11-11 16:39:33	Afternoon	Weekday	2
11	3005	Alfajores	2016-11-12 13:08:04	Afternoon	Weekend	2

```
Query executed successfully. | LAPTOP-KPPUFOIL\SQLEXPRESS ... | LAPTOP-KPPUFOIL\أحمد... | Bakery_store | 00:00:42 | 1,620 rows
```

Scrub data (S)

-We delete duplicate records:

We used a SQL function called (CTE) and the ROW_NUMBER() function to assign a unique number to each row in the bakery table based on the values in the [Transaction No], Items, DateTime, and [DayPart] columns. Then, it deletes any rows with a row number greater than 1, effectively removing duplicate rows from the table.



```sql
--Delete duplicate rows

WITH CTE AS (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY [TransactionNo],
    Items, DateTime, [DayPart] ORDER BY [TransactionNo]) AS RN
    FROM bakery
)
DELETE FROM CTE
WHERE RN > 1;

--Checking the number of rows after deleting duplicate values

select * from bakery
```

Results | Messages

TransactionNo	Items	DateTime	Daypart	DayType
1	Bread	2016-10-30 09:58:11	Morning	Weekend
2	Scandinavian	2016-10-30 10:05:34	Morning	Weekend
3	Cookies	2016-10-30 10:07:57	Morning	Weekend

```
Query executed successfully.   LAPTOP-KPPUFOIL\SQLEXPRESS ...   LAPTOP-KPPUFOIL\أحمد...   Bakery_store   00:00:00   18,887 rows
```

18,887 unique rows checked.

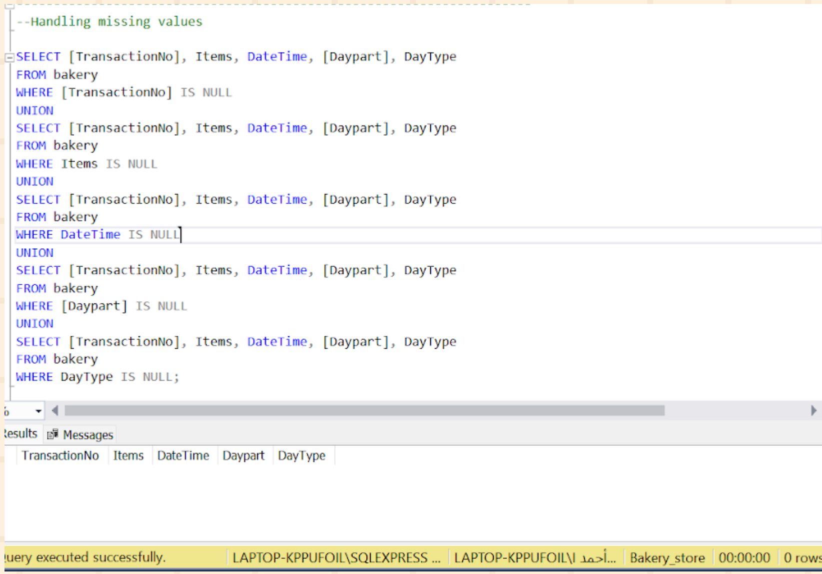
Then we move on to the second step in the data cleaning process, which is to process the missing values

Scrub data (S)

2) Handling missing values: We checked for missing values and noticed that there weren't any missing values.

Instead of querying for the missing values for each column individually, we used the IS NULL operator and the UNION operator to retrieve the missing values in all columns of the "bakery" table.

Then we move on to the third stage in the cleaning process, which is the conversion of data types



```
--Handling missing values

SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE [TransactionNo] IS NULL
UNION
SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE Items IS NULL
UNION
SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE DateTime IS NULL
UNION
SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE [Daypart] IS NULL
UNION
SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE DayType IS NULL;
```

TransactionNo	Items	DateTime	Daypart	DayType

```
uery executed successfully.   LAPTOP-KPPUFOIL\SQLEXPRESS ...   LAPTOP-KPPUFOIL\أحمد...   Bakery_store   00:00:00   0 rows
```

Scrub data (S)

3) Convert data types:

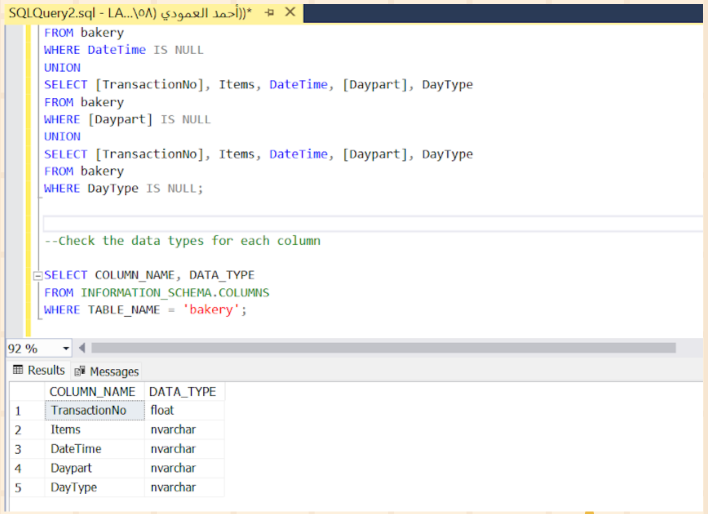
Convert data types to any variables as needed to ensure they are suitable for analysis.

We checked the data type for each column and found an error

We note that the TransactionNo column must be an integer, the Items column must be VARCHAR, the DateTime column must be DATETIME, and the Daypart and DayType columns must be VARCHAR.

Then we fixed the data type of the columns in the spreadsheet and checked and everything is ok.

Then we move on to the fourth step in the data cleaning process, which is to correct the inconsistent format



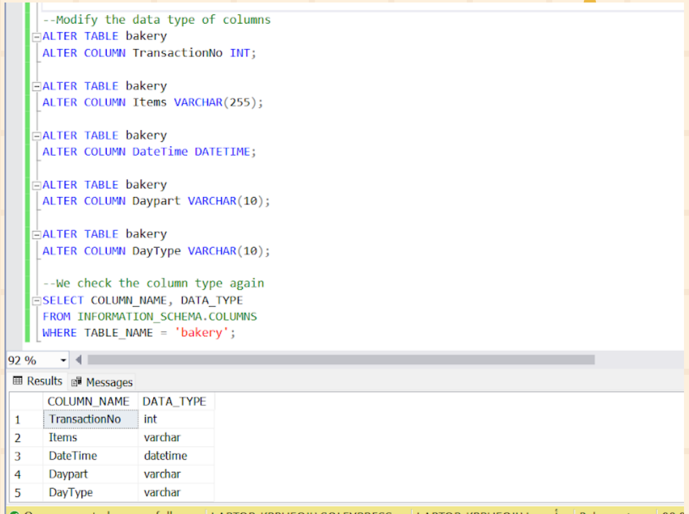
SQLQuery2.sql - LA...\٥٨ (أحمد العمودي))*

```
FROM bakery
WHERE DateTime IS NULL
UNION
SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE [Daypart] IS NULL
UNION
SELECT [TransactionNo], Items, DateTime, [Daypart], DayType
FROM bakery
WHERE DayType IS NULL;

--Check the data types for each column

SELECT COLUMN_NAME, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'bakery';
```

92 %

Results | Messages

	COLUMN_NAME	DATA_TYPE
1	TransactionNo	float
2	Items	nvarchar
3	DateTime	nvarchar
4	Daypart	nvarchar
5	DayType	nvarchar



```
--Modify the data type of columns
ALTER TABLE bakery
ALTER COLUMN TransactionNo INT;

ALTER TABLE bakery
ALTER COLUMN Items VARCHAR(255);

ALTER TABLE bakery
ALTER COLUMN DateTime DATETIME;

ALTER TABLE bakery
ALTER COLUMN Daypart VARCHAR(10);

ALTER TABLE bakery
ALTER COLUMN DayType VARCHAR(10);

--We check the column type again
SELECT COLUMN_NAME, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'bakery';
```

92 %

Results | Messages

	COLUMN_NAME	DATA_TYPE
1	TransactionNo	int
2	Items	varchar
3	DateTime	datetime
4	Daypart	varchar
5	DayType	varchar

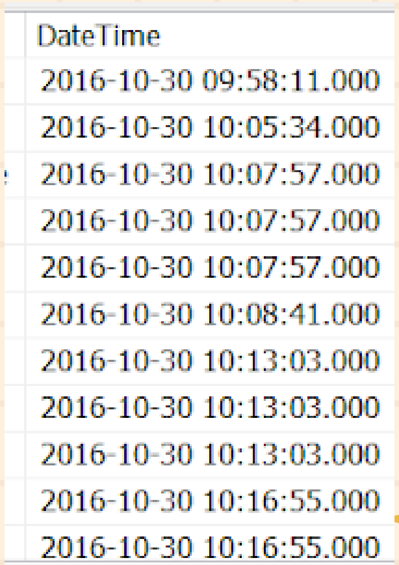
Scrub data (S)

4) Correct inconsistent formatting:

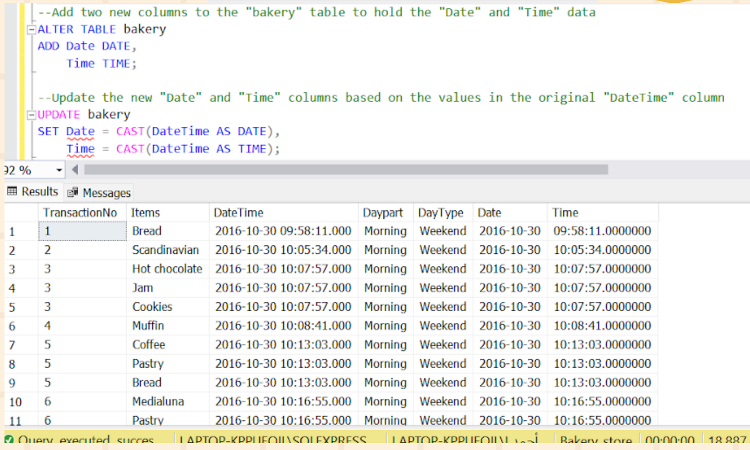
We notice that the date and time column are merged together and we want to separate them.

- We made queries that create the new “Date” and “Time” columns in the Bakery table, and update them with the appropriate values from the original “DateTime” column.

And then we moved on to the stage of removing unnecessary columns



DateTime
2016-10-30 09:58:11.000
2016-10-30 10:05:34.000
2016-10-30 10:07:57.000
2016-10-30 10:07:57.000
2016-10-30 10:07:57.000
2016-10-30 10:08:41.000
2016-10-30 10:13:03.000
2016-10-30 10:13:03.000
2016-10-30 10:13:03.000
2016-10-30 10:16:55.000
2016-10-30 10:16:55.000



```
--Add two new columns to the "bakery" table to hold the "Date" and "Time" data
ALTER TABLE bakery
ADD Date DATE,
    Time TIME;

--Update the new "Date" and "Time" columns based on the values in the original "DateTime" column
UPDATE bakery
SET Date = CAST(DateTime AS DATE),
    Time = CAST(DateTime AS TIME);
```

	TransactionNo	Items	DateTime	Daypart	DayType	Date	Time
1	1	Bread	2016-10-30 09:58:11.000	Morning	Weekend	2016-10-30	09:58:11.0000000
2	2	Scandinavian	2016-10-30 10:05:34.000	Morning	Weekend	2016-10-30	10:05:34.0000000
3	3	Hot chocolate	2016-10-30 10:07:57.000	Morning	Weekend	2016-10-30	10:07:57.0000000
4	3	Jam	2016-10-30 10:07:57.000	Morning	Weekend	2016-10-30	10:07:57.0000000
5	3	Cookies	2016-10-30 10:07:57.000	Morning	Weekend	2016-10-30	10:07:57.0000000
6	4	Muffin	2016-10-30 10:08:41.000	Morning	Weekend	2016-10-30	10:08:41.0000000
7	5	Coffee	2016-10-30 10:13:03.000	Morning	Weekend	2016-10-30	10:13:03.0000000
8	5	Pastry	2016-10-30 10:13:03.000	Morning	Weekend	2016-10-30	10:13:03.0000000
9	5	Bread	2016-10-30 10:13:03.000	Morning	Weekend	2016-10-30	10:13:03.0000000
10	6	Medialuna	2016-10-30 10:16:55.000	Morning	Weekend	2016-10-30	10:16:55.0000000
11	6	Pastry	2016-10-30 10:16:55.000	Morning	Weekend	2016-10-30	10:16:55.0000000

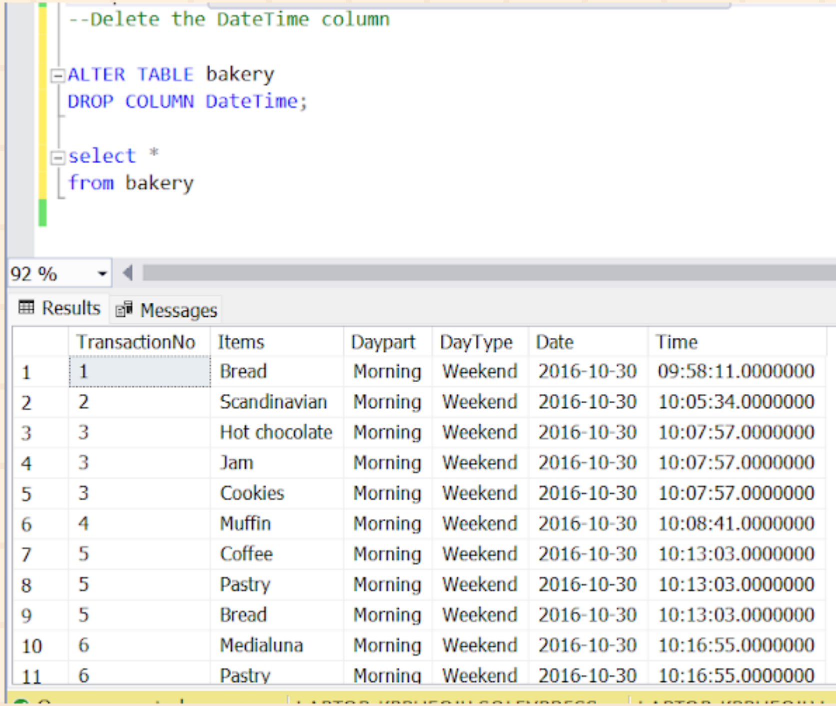
```
Query executed succes | LAPTOP-KPPUEOII\SQLEXPRESS | LAPTOP-KPPUEOII\ أحمد | Bakery store | 00:00:00 | 18,887
```

Scrub data (S)

5) Remove irrelevant data:

We remove the “DateTime” column because it is not important

Then we move on to the second process of the OSEMN framework, which is Explore (E).



```
--Delete the DateTime column

ALTER TABLE bakery
DROP COLUMN DateTime;

select *
from bakery
```

92 %

Results | Messages

	TransactionNo	Items	Daypart	DayType	Date	Time
1	1	Bread	Morning	Weekend	2016-10-30	09:58:11.0000000
2	2	Scandinavian	Morning	Weekend	2016-10-30	10:05:34.0000000
3	3	Hot chocolate	Morning	Weekend	2016-10-30	10:07:57.0000000
4	3	Jam	Morning	Weekend	2016-10-30	10:07:57.0000000
5	3	Cookies	Morning	Weekend	2016-10-30	10:07:57.0000000
6	4	Muffin	Morning	Weekend	2016-10-30	10:08:41.0000000
7	5	Coffee	Morning	Weekend	2016-10-30	10:13:03.0000000
8	5	Pastry	Morning	Weekend	2016-10-30	10:13:03.0000000
9	5	Bread	Morning	Weekend	2016-10-30	10:13:03.0000000
10	6	Medialuna	Morning	Weekend	2016-10-30	10:16:55.0000000
11	6	Pastry	Morning	Weekend	2016-10-30	10:16:55.0000000

Explore data (E)

exploration phase (E).

To explore datasets, these three steps must be followed:

First, examine the variable distributions.

Second, test the changing relationships.

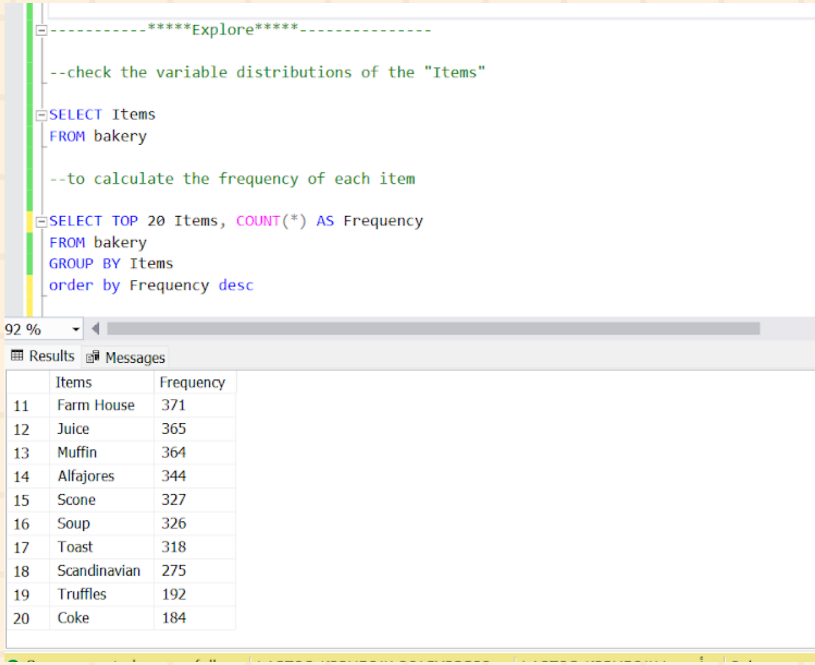
Third, do the statistical tests.

Explore data (E)

1) Examine the variable distributions

To check the variable distributions of the "Items" variable in a bakery dataset, you can use SQL Server functions and tools to analyze the frequency and distribution of each item sold.

- We made a query that will return a list of items and their corresponding occurrences in the bakery dataset. We selected only the top 20 most frequent elements.



```
-----------*****Explore*****--------------

--check the variable distributions of the "Items"

SELECT Items
FROM bakery

--to calculate the frequency of each item

SELECT TOP 20 Items, COUNT(*) AS Frequency
FROM bakery
GROUP BY Items
order by Frequency desc
```

```
    Items         Frequency
11  Farm House    371
12  Juice         365
13  Muffin        364
14  Alfajores     344
15  Scone         327
16  Soup          326
17  Toast         318
18  Scandinavian  275
19  Truffles      192
20  Coke          184
```

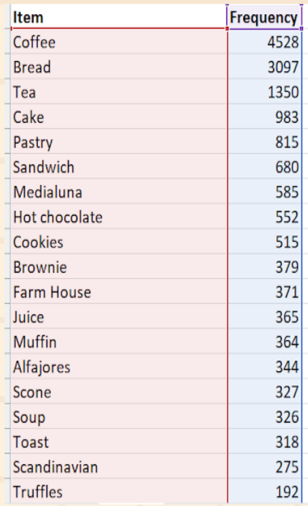
Explore data (E)

1) Examine the variable distributions

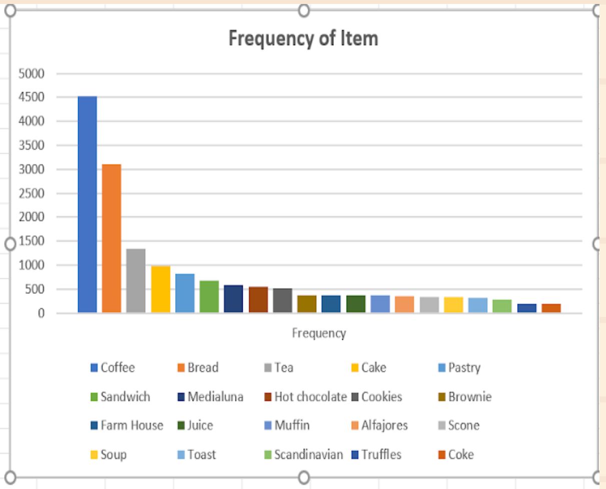
After that, we exported this data to the Excel program in order to create a graph to show us which items were purchased most frequently.

Through the bar chart we got an insight into which items are the most popular and which items are not so popular.
We found that the following three items are the most popular: coffee, bread and tea

Then we moved to the second step in the data exploration process, which is testing the variable relationships



Item	Frequency
Coffee	4528
Bread	3097
Tea	1350
Cake	983
Pastry	815
Sandwich	680
Medialuna	585
Hot chocolate	552
Cookies	515
Brownie	379
Farm House	371
Juice	365
Muffin	364
Alfajores	344
Scone	327
Soup	326
Toast	318
Scandinavian	275
Truffles	192



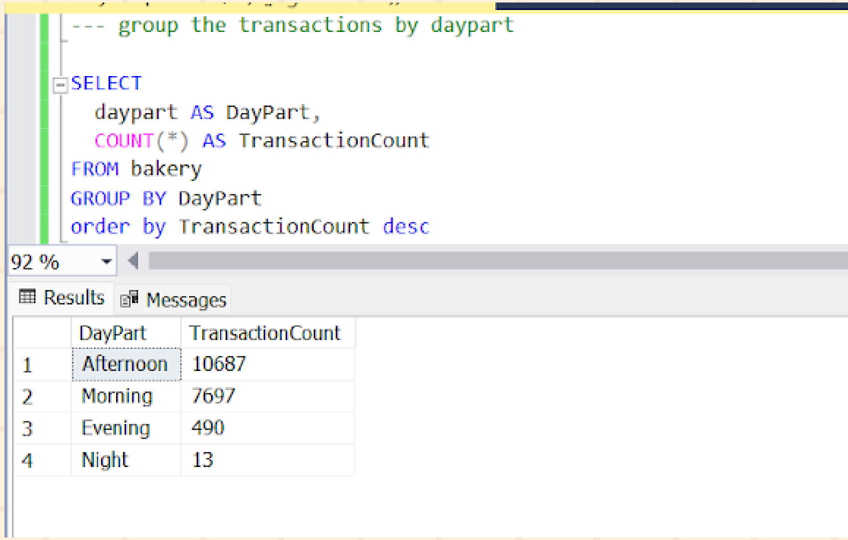
Explore data (E)

2) Variable relationship test

We looked at the relationship between the “daypart” variable and “TransactionNo” in order to see which part of the day is crowded.

As we will notice that any part of the day in which the transaction takes place is crowded

- We made a query that aggregates these transactions by column "DayPart" and counts the number of transactions in each group using the COUNT function



```
--- group the transactions by daypart

SELECT
  daypart AS DayPart,
  COUNT(*) AS TransactionCount
FROM bakery
GROUP BY DayPart
order by TransactionCount desc
```

92 %

Results | Messages

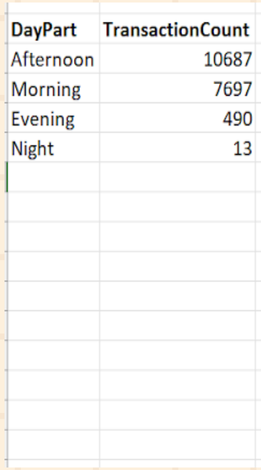
	DayPart	TransactionCount
1	Afternoon	10687
2	Morning	7697
3	Evening	490
4	Night	13

Explore data (E)

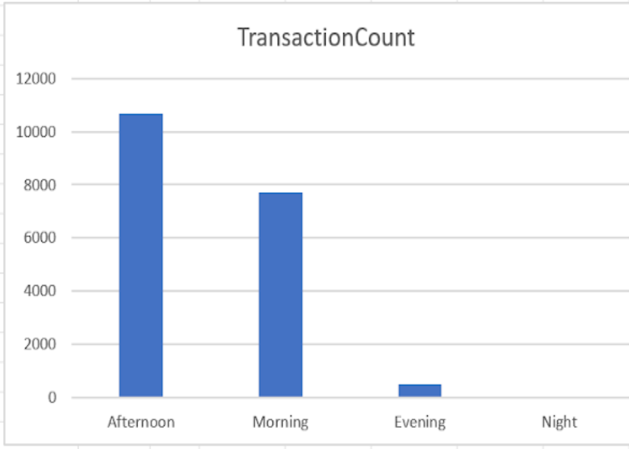
2) Variable relationship test

We exported the data of the result of this query to Excel to create a bar chart

Based on these results, the bakery may wish to consider increasing the number of employees during the afternoon and morning hours to handle the higher volume of transactions during those times. In addition, the bakery may want to consider offering promotions or discounts during the less busy evening and night hours to attract more customers during those times.



DayPart	TransactionCount
Afternoon	10687
Morning	7697
Evening	490
Night	13



Explore data (E)

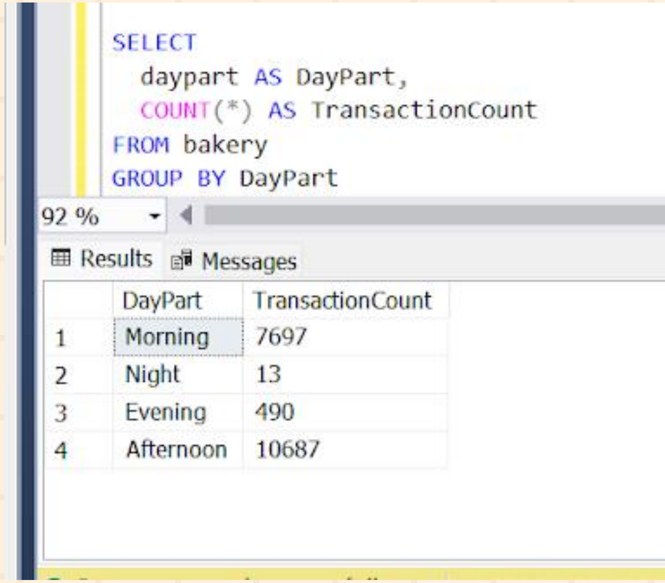
3) Carry out statistical tests

We are going to run a statistical test to see if our conclusion about the parts of the day that there were transactions is just the result of chance first.

We tested chi-square for independence in SQL Server, by using the following steps:

1. Create a contingency table: We need to create a contingency table showing the observed frequency of transactions for each period of the day. By creating use the following SQL query to create a contingency table

This query groups transactions by "DayPart" column and counts the number of transactions in each group using the COUNT function.



```
SELECT
   daypart AS DayPart,
   COUNT(*) AS TransactionCount
FROM bakery
GROUP BY DayPart
```

92 %

Results | Messages

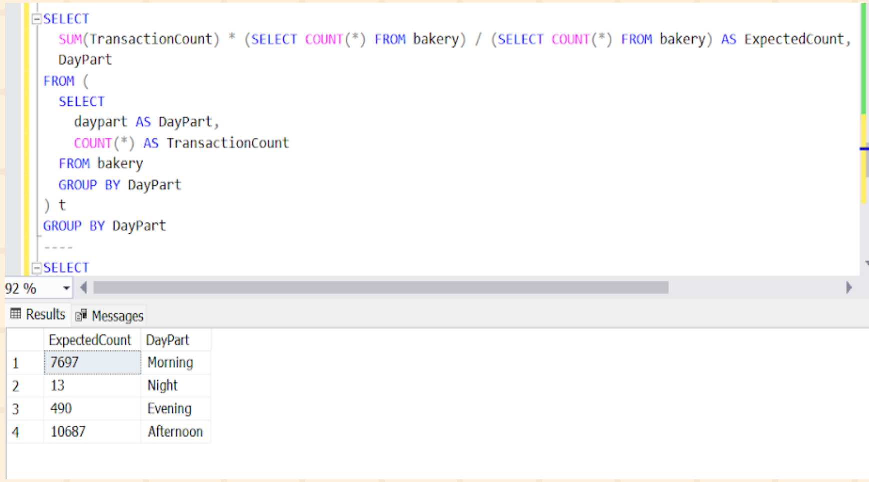
	DayPart	TransactionCount
1	Morning	7697
2	Night	13
3	Evening	490
4	Afternoon	10687

Explore data (E)

3) Carry out statistical tests

2. Calculating Expected Frequencies: We need to calculate expected occurrences for each period of the day based on the total number of transactions and the percentage of transactions that occur during that day.

We make a query that first calculates the total number of transactions in the bakery table and then computes the expected number of transactions for each period of the day based on the percentage of transactions that occur during that day.



```
SELECT
  SUM(TransactionCount) * (SELECT COUNT(*) FROM bakery) / (SELECT COUNT(*) FROM bakery) AS ExpectedCount,
  DayPart
FROM (
  SELECT
    daypart AS DayPart,
    COUNT(*) AS TransactionCount
  FROM bakery
  GROUP BY DayPart
) t
GROUP BY DayPart
----
SELECT
```

92 %

Results | Messages

	ExpectedCount	DayPart
1	7697	Morning
2	13	Night
3	490	Evening
4	10687	Afternoon

Explore data (E)

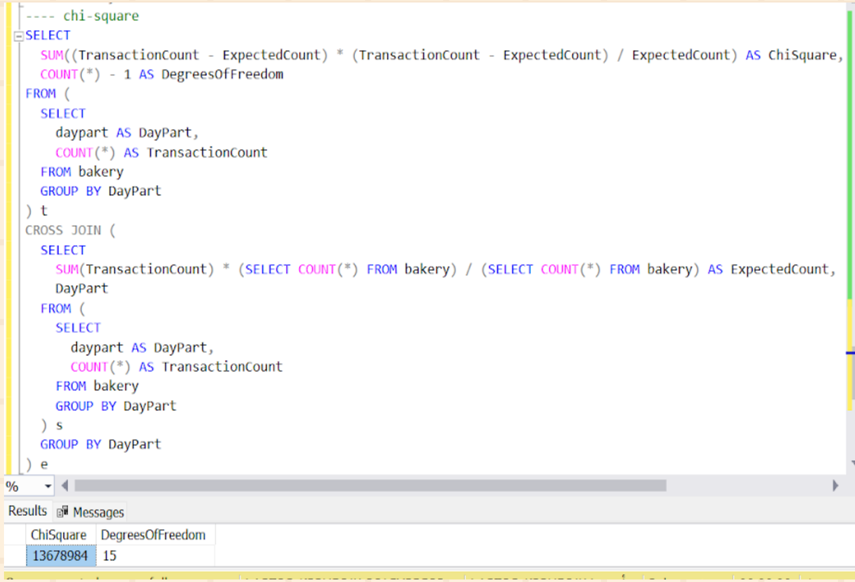
3) Carry out statistical tests

3. Calculating the chi-square statistic:

This query first computes the observed frequencies of the transactions for each part of the day (using the same subquery as in Step 1) and then relates this to the expected frequencies (using the subquery from Step 2). It then calculates a chi-square statistic using the formula X2 = Σ (O - E)2 / E where O is the observed frequency and E is the expected frequency.

To find the critical value of the chi-square independence test, you need to determine the significance level (alpha) and degrees of freedom. The degrees of freedom for this test are calculated as (number of rows - 1) * (number of columns - 1), which in this case is (4 - 1) * (2 - 1) = 3

Assuming a significance level of 0.05 (that is, alpha = 0.05), we can find the critical value for a chi-square distribution table with 15 degrees of freedom. When you look up this value in a chi-square distribution table, you can find that the critical value is approximately 24.996.



```
---- chi-square
SELECT
  SUM((TransactionCount - ExpectedCount) * (TransactionCount - ExpectedCount) / ExpectedCount) AS ChiSquare,
  COUNT(*) - 1 AS DegreesOfFreedom
FROM (
  SELECT
    daypart AS DayPart,
    COUNT(*) AS TransactionCount
  FROM bakery
  GROUP BY DayPart
) t
CROSS JOIN (
  SELECT
    SUM(TransactionCount) * (SELECT COUNT(*) FROM bakery) / (SELECT COUNT(*) FROM bakery) AS ExpectedCount,
    DayPart
  FROM (
    SELECT
      daypart AS DayPart,
      COUNT(*) AS TransactionCount
    FROM bakery
    GROUP BY DayPart
  ) s
  GROUP BY DayPart
) e
```

Results Messages

ChiSquare	DegreesOfFreedom
13678984	15

Explore data (E)

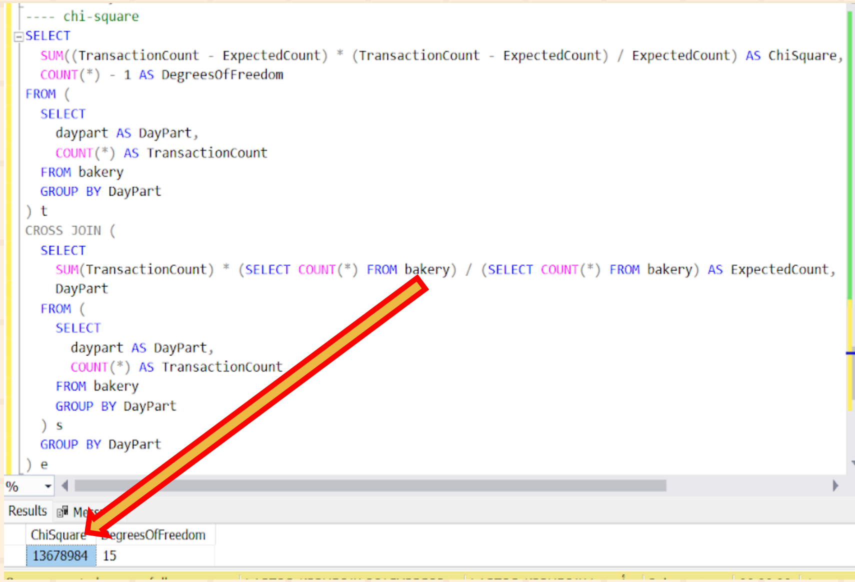
3) Carry out statistical tests

3. Calculating the chi-square statistic:

Since the calculated chi-square statistic of **13,678,984** is much larger than the critical value of 24,996, you can reject the null hypothesis and conclude that **there is a significant correlation between Part of Day and the number of transactions.**

This means that the distribution of transactions across different parts of the day is not simply a result of chance, but is statistically significant instead.

Then we move on to the Modeling stage of the OSEMN framework



```
---- chi-square
SELECT
  SUM((TransactionCount - ExpectedCount) * (TransactionCount - ExpectedCount) / ExpectedCount) AS ChiSquare,
  COUNT(*) - 1 AS DegreesOfFreedom
FROM (
  SELECT
    daypart AS DayPart,
    COUNT(*) AS TransactionCount
  FROM bakery
  GROUP BY DayPart
) t
CROSS JOIN (
  SELECT
    SUM(TransactionCount) * (SELECT COUNT(*) FROM bakery) / (SELECT COUNT(*) FROM bakery) AS ExpectedCount,
    DayPart
  FROM (
    SELECT
      daypart AS DayPart,
      COUNT(*) AS TransactionCount
    FROM bakery
    GROUP BY DayPart
  ) s
  GROUP BY DayPart
) e
```

```
ChiSquare   DegreesOfFreedom
13678984    15
```

Modeling (M)

Depending on the questions we need to answer, and since we are looking to perform a market basket analysis to identify patterns and correlations among items purchased. Therefore, the appropriate model to use at this point would be the **association rule mining model**.

We used the **Python** programming language at this stage, after exporting the data from SQL server, we loaded it into the Jupyter Notebook program

Modeling (M)

Correlation rule mining is a popular technique used in market basket analysis to identify groups of recurring items and association rules between items. It can help us answer questions such as which items are purchased most frequently together, how often customers buy certain items together, and we can sell or cross-sell certain items based on customers' buying patterns.

Once we create association rules, we can use them to answer our other questions such as which products or items are most popular for sale during certain times of the day or week, and how we can improve product placement in stores to increase sales.

Overall, the association rule mining model is the right one to use at this point to perform a market basket analysis and gain insight into the buying patterns of your customers.

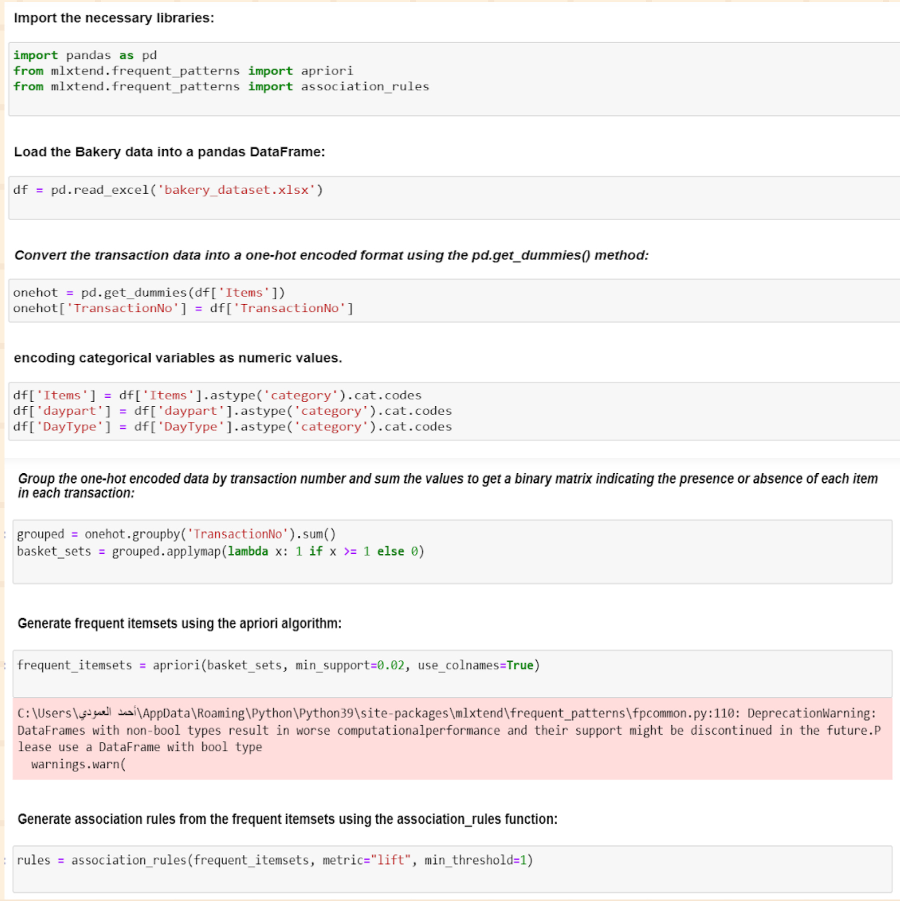
Modeling (M)

We first create association rules:

To use the association rule mining model in our project, we have done the following steps:

First we imported the necessary libraries:
Next we loaded the bakery data into the pandas data frame.
Hence we processed the data as needed. We coded categorical variables as numeric values.
Next we converted the bakery data to a one-hot encoded format using the pd.get_dummies() method.

Hence we group the hot encoded data by transaction number and sum the values to obtain a binary matrix indicating the presence or absence of each item in each transaction:
Next we created recursive element sets using the apriori algorithm:
Hence we create association rules from groups of repeating elements using association_rules function:



Import the necessary libraries:

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

Load the Bakery data into a pandas DataFrame:

```
df = pd.read_excel('bakery_dataset.xlsx')
```

Convert the transaction data into a one-hot encoded format using the pd.get_dummies() method:

```
onehot = pd.get_dummies(df['Items'])
onehot['TransactionNo'] = df['TransactionNo']
```

encoding categorical variables as numeric values.

```
df['Items'] = df['Items'].astype('category').cat.codes
df['daypart'] = df['daypart'].astype('category').cat.codes
df['DayType'] = df['DayType'].astype('category').cat.codes
```

Group the one-hot encoded data by transaction number and sum the values to get a binary matrix indicating the presence or absence of each item in each transaction:

```
grouped = onehot.groupby('TransactionNo').sum()
basket_sets = grouped.applymap(lambda x: 1 if x >= 1 else 0)
```

Generate frequent itemsets using the apriori algorithm:

```
frequent_itemsets = apriori(basket_sets, min_support=0.02, use_colnames=True)
```

```
C:\Users\أحمد العمودي\AppData\Roaming\Python\Python39\site-packages\mlxtend\frequent_patterns\fpcommon.py:110: DeprecationWarning:
DataFrames with non-bool types result in worse computationalperformance and their support might be discontinued in the future.P
lease use a DataFrame with bool type
  warnings.warn(
```

Generate association rules from the frequent itemsets using the association_rules function:

```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
```

Modeling (M)

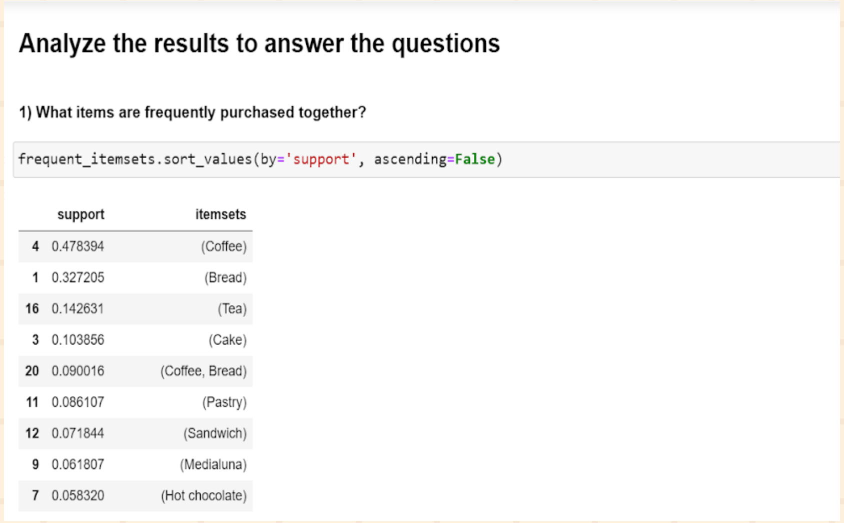
Analyze the results of the association rules mining model and answer the questions

We can use the sets of repeating elements and association rules created in the previous steps to analyze the results of the association rule mining model and answer the questions in our project

Here are some ways to analyze the results and answer the questions:

1) What items are frequently purchased together?

We can use a DataFrame to define the most common sets of elements in the data, sorted by support. For example, we can sort the DataFrame by support column in descending order to see the most common element groups:



Analyze the results to answer the questions

1) What items are frequently purchased together?

```
frequent_itemsets.sort_values(by='support', ascending=False)
```

	support	itemsets
4	0.478394	(Coffee)
1	0.327205	(Bread)
16	0.142631	(Tea)
3	0.103856	(Cake)
20	0.090016	(Coffee, Bread)
11	0.086107	(Pastry)
12	0.071844	(Sandwich)
9	0.061807	(Medialuna)
7	0.058320	(Hot chocolate)

Modeling (M)

Analyze the results of the association rules mining model and answer the questions

1) What items are frequently purchased together?

We displayed item groups in descending order of support, which represents the frequency with which the item group appears in transactions.
Each row in the result represents an element set, which is a collection of elements that appear together frequently in transactions. The support measure refers to the portion of the moves that contain the set of items, which reflects the frequency of the set of items in the moves.

This score provides insight into the items that are most frequently purchased at the bakery, as well as the groups of items that are most frequently purchased together. This information can be used to direct the bakery's product placement and marketing strategies, such as putting together related items, creating package deals, or promoting certain items to increase sales.

Modeling (M)

Analyze the results of the association rules mining model and answer the questions

1) What items are frequently purchased together?

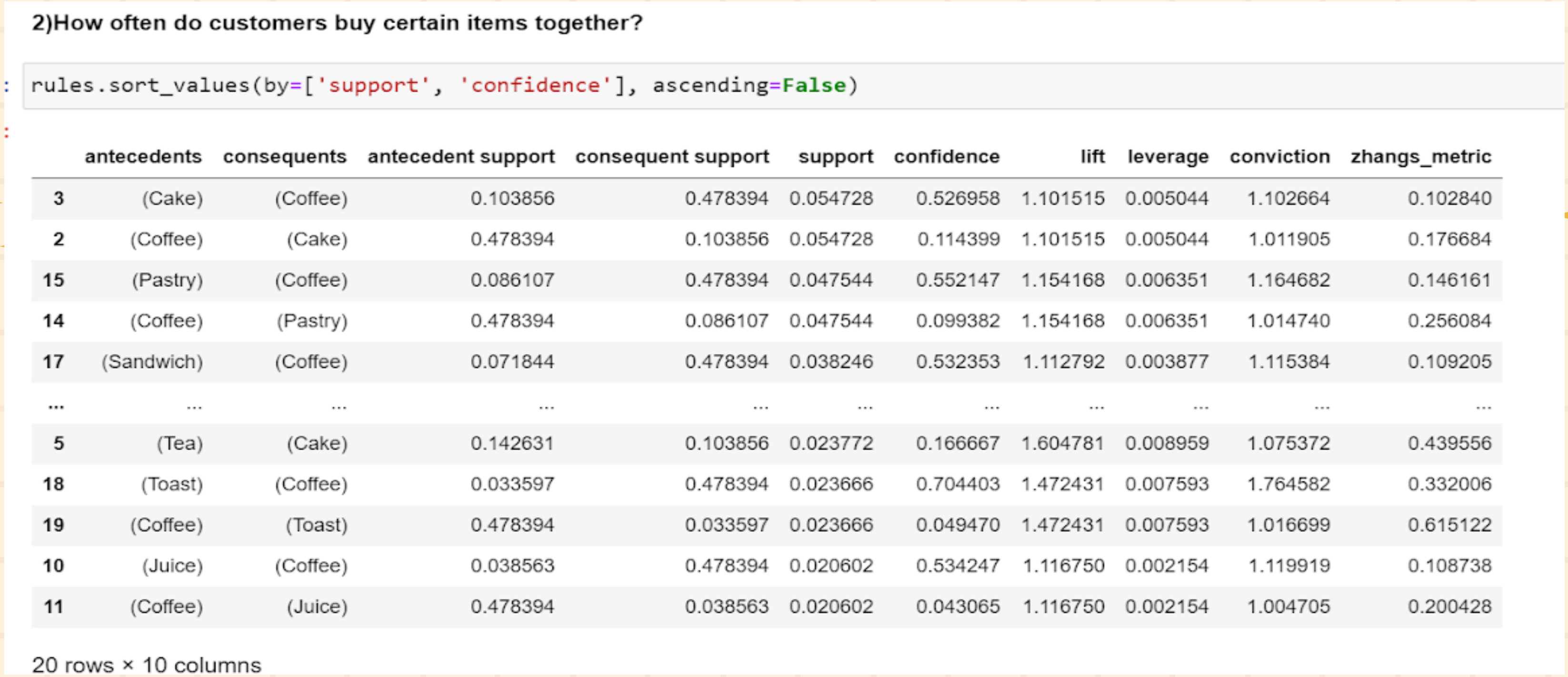
For example, the first row of the result shows that "coffee" appears in 47.84% of all transactions, meaning it is the most frequently purchased item. The second row shows that "bread" appears in 32.72% of all transactions, making it the second most frequently purchased item. The third row shows that the word "Tea" appears in 14.26% of all transactions, making it the third most frequently purchased item.

The remaining rows show the groups of elements that appear together most frequently in transactions, sorted by support in descending order. For example, the item combination ("Coffee", "Bread") appears in 9.00% of all transactions, making it the most purchased combination of two items. The next most purchased items are (“tea” and “cake”), (“coffee” and “cake”) and (“coffee” and “pastries”), appearing in 2.36%, 2.37% and 4.95% of all transactions, respectively.

Modeling (M)

Analyze the results of the association rules mining model and answer the questions

2) How often do customers buy certain items together?



2)How often do customers buy certain items together?

```
rules.sort_values(by=['support', 'confidence'], ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
3	(Cake)	(Coffee)	0.103856	0.478394	0.054728	0.526958	1.101515	0.005044	1.102664	0.102840
2	(Coffee)	(Cake)	0.478394	0.103856	0.054728	0.114399	1.101515	0.005044	1.011905	0.176684
15	(Pastry)	(Coffee)	0.086107	0.478394	0.047544	0.552147	1.154168	0.006351	1.164682	0.146161
14	(Coffee)	(Pastry)	0.478394	0.086107	0.047544	0.099382	1.154168	0.006351	1.014740	0.256084
17	(Sandwich)	(Coffee)	0.071844	0.478394	0.038246	0.532353	1.112792	0.003877	1.115384	0.109205
...	...	...	...	...	...	...	...	...	...	...
5	(Tea)	(Cake)	0.142631	0.103856	0.023772	0.166667	1.604781	0.008959	1.075372	0.439556
18	(Toast)	(Coffee)	0.033597	0.478394	0.023666	0.704403	1.472431	0.007593	1.764582	0.332006
19	(Coffee)	(Toast)	0.478394	0.033597	0.023666	0.049470	1.472431	0.007593	1.016699	0.615122
10	(Juice)	(Coffee)	0.038563	0.478394	0.020602	0.534247	1.116750	0.002154	1.119919	0.108738
11	(Coffee)	(Juice)	0.478394	0.038563	0.020602	0.043065	1.116750	0.002154	1.004705	0.200428

20 rows × 10 columns

Modeling (M)

Analyze the results of the association rules mining model and answer the questions

2) How often do customers buy certain items together?

Each row in the score represents an association rule, consisting of an antecedent (left side) and a follower (the right side), along with different measures describing the strength and significance of the association.

Here is a breakdown of the columns in the result:

Antecedent: The set of elements that appear in the prefix of the rule, separated by commas.
Consequent: The set of elements that appear following the rule, separated by commas.
Support: Transaction pane that contains a prefix and a successor.
Confidence: The part of transactions that contains a precedent and also contains a consequence.
Lift: the ratio of the observed support to the expected support, assuming that the antecedent and consequent are independent.
Leverage: The difference between the observed support and the expected support, assuming independence.
Conviction: The ratio of predicted confidence (assuming independence) to observed confidence.
Antecedent support: Part of the transaction that contains an antecedent.
Consequent support: The part of transactions that contains consequences.

Modeling (M)

Analyze the results of the association rules mining model and answer the questions

2) How often do customers buy certain items together?

For example, let's look at the first row of the result:

Anticedent A:	3 (cake)
Consequent:	2 (coffee)
Support:	0.103856
Confidence:	0.478394
A Lift:	1.101515
Leverage:	0.005044
Conviction:	1.102664
Antecedent support:	0.216196
Consequent support:	0.478394

This rule indicates that customers who buy “cake” are likely to also buy “coffee”, with a support of 0.103856, which means that this group of items appears in 10.39% of all transactions. The confidence of this rule is 0.478394, which means that 47.84% of transactions containing “cake” also contain “coffee”. The base lift is 1.101515, **which means that the observed support for this element group is 1.101515 times higher than what would be expected if “Cake” and “Coffee” were purchased independently**. Other measures provide additional information about the strength and significance of the association.

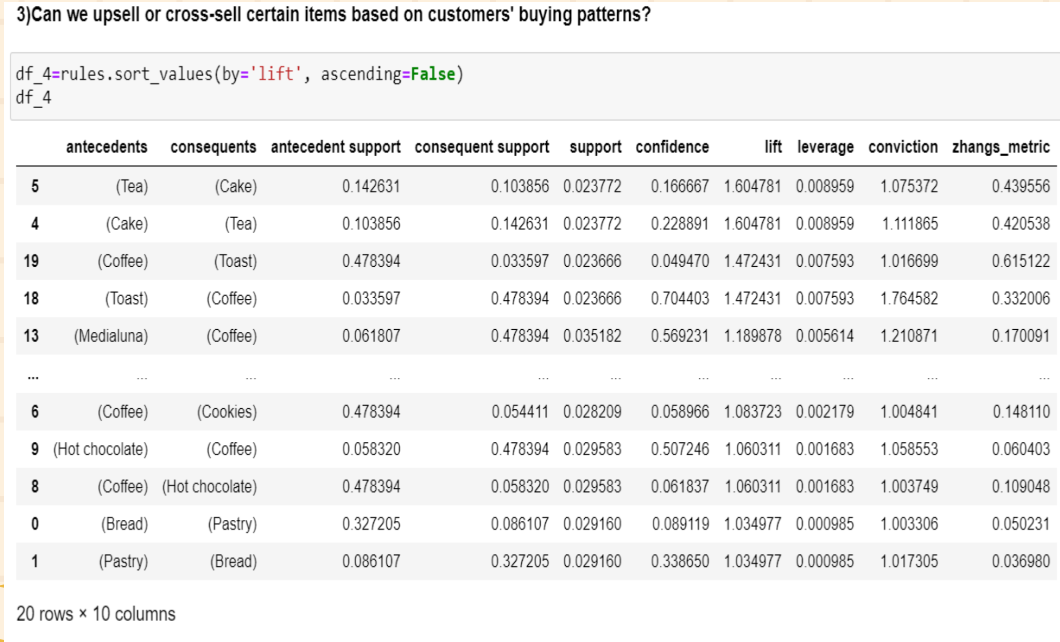
Modeling (M)

Analyze the results of the association rules mining model and answer the questions

3) Can we upsell or cross-sell certain items based on customer buying patterns?

The displayed result is a list of correlation rules generated by the market basket analysis of bakery data. Each rule consists of an antecedent (items most likely to be purchased together),

- a consequent (item(s) likely to be purchased when an antecedent is purchased), and some statistical measure of the strength of association between the antecedent and consequent.



3)Can we upsell or cross-sell certain items based on customers' buying patterns?

```
df_4=rules.sort_values(by='lift', ascending=False)
df_4
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
5	(Tea)	(Cake)	0.142631	0.103856	0.023772	0.166667	1.604781	0.008959	1.075372	0.439556
4	(Cake)	(Tea)	0.103856	0.142631	0.023772	0.228891	1.604781	0.008959	1.111865	0.420538
19	(Coffee)	(Toast)	0.478394	0.033597	0.023666	0.049470	1.472431	0.007593	1.016699	0.615122
18	(Toast)	(Coffee)	0.033597	0.478394	0.023666	0.704403	1.472431	0.007593	1.764582	0.332006
13	(Medialuna)	(Coffee)	0.061807	0.478394	0.035182	0.569231	1.189878	0.005614	1.210871	0.170091
...	...	...	...	...	...	...	...	...	...	...
6	(Coffee)	(Cookies)	0.478394	0.054411	0.028209	0.058966	1.083723	0.002179	1.004841	0.148110
9	(Hot chocolate)	(Coffee)	0.058320	0.478394	0.029583	0.507246	1.060311	0.001683	1.058553	0.060403
8	(Coffee)	(Hot chocolate)	0.478394	0.058320	0.029583	0.061837	1.060311	0.001683	1.003749	0.109048
0	(Bread)	(Pastry)	0.327205	0.086107	0.029160	0.089119	1.034977	0.000985	1.003306	0.050231
1	(Pastry)	(Bread)	0.086107	0.327205	0.029160	0.338650	1.034977	0.000985	1.017305	0.036980

20 rows × 10 columns

Modeling (M)

Analyze the results of the association rules mining model and answer the questions

3) Can we upsell or cross-sell certain items based on customer buying patterns?

We sorted the data to create this grammar score by the value of 'lift', which is a measure of how likely the outcome is to buy an antecedent, compared to the overall probability of buying a consequential purchase. In general, a higher lift value indicates a stronger association between antecedent and consequent.

Looking at the higher bases in the score**, we can see that the 'Tea' element is closely related to the 'Cake' element (and vice versa), with a lift value of 1.60. This indicates that customers who buy tea are also likely to buy cake, and vice versa**. This information can be used to upsell or cross-sell those items to customers who already purchase one.

Another strong correlation in the upper bases is between the items "Coffee" and "Toast", with a lift value of 1.47. This indicates that customers who buy coffee are also more likely to buy toast, and vice versa. This information can be used to upsell or cross-sell those items to customers who already purchase one.

Other rules in the score may also be useful in upselling or cross selling, for example the “Hot chocolate” -> “Coffee” rule suggests that customers who buy hot chocolate are more likely to buy coffee, which can be useful for promoting sales of coffee during months winter. The "Bread" -> "Pastry" rule suggests that customers who buy bread are also more likely to buy pastries, which can be useful for promoting pastry sales during breakfast hours.

iNterpret (N)

The goal of a market basket analysis project is to use the insights gained from the analysis to enhance customer satisfaction, improve product offerings, and improve sales and marketing strategies. To achieve this goal, we aim to answer questions such as which items are commonly purchased together,

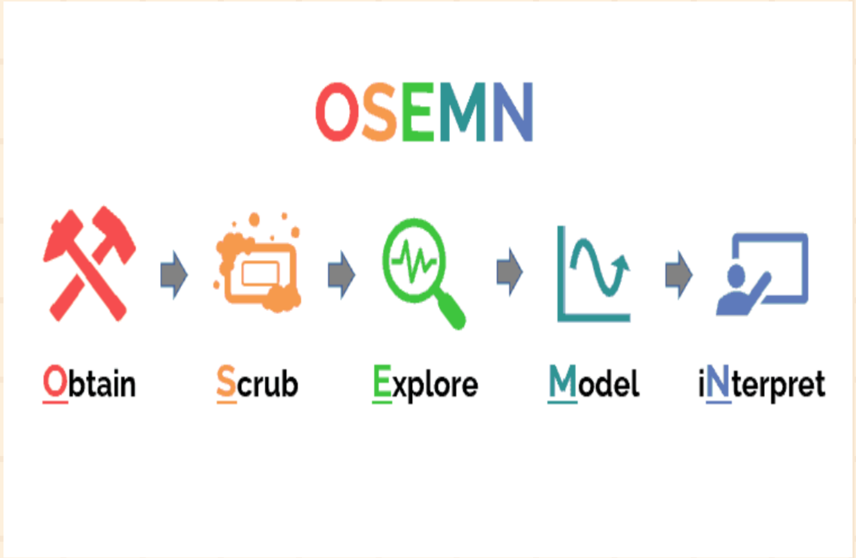
how frequently certain items are purchased together by customers, and whether certain items can be sold or cross-sold based on customers' buying patterns. By providing answers to these questions, market basket analysis can help companies identify potential areas for improvement in their sales and marketing strategies, product offerings, and overall customer satisfaction.

iNterpret (N)

We have used the **OSEMN** framework in our project.
And then we entered into the first stage,
which is to **obtain the data**:

The dataset for Market Basket Analysis was obtained from **Kaggle** and contains transaction data for the 'Breadbasket' bakery in Edinburgh. It includes over 9,000 transactions and 20,507 entries with four columns - transaction number, items, date and time, and day part.

After obtaining the data, we uploaded it to the SQL server program to perform steps to clean and explore the data.



iNterpret (N)

We **first cleaned the data** with the goal of ensuring that the data set is accurate, complete, consistent, and ready for analysis.

We have implemented the steps to clean the data using SQL server, which are:

1-Remove Duplicates: We have identified any duplicate records in the dataset and removed them.

2-Handling Missing Values: We checked for missing values and noticed that there weren't any missing values.

3- Converting data types: we checked the data type for each column and found an error and fixed it.

4. Correct inconsistent formatting. We notice that the date and time column are merged with each other and we have separated them.

5- We removed irrelevant data.

iNterpret (N)

Then we moved to the **data exploration** stage:
To explore the datasets, we followed these three steps of data exploration.

First we examined the variable distributions:
of the "Items" variable in the bakery dataset, using SQL Server functions and tools to analyze the frequency and distribution of each item sold. Through an SQL query, we view a list of items and their corresponding occurrences in the bakery dataset. We selected only the top 20 most frequent elements.

Then we exported the data of this result to the Excel program in order to make a graph to show us which items were purchased frequently.

We made a bar chart to get an insight into which items are the most popular and which are not so popular.
We found that the following three **items are the most popular: coffee, bread and tea.**

iNterpret (N)

Then we proceeded to the **second step in the exploration phase, which is testing the variable relationships**:

We tested the relationship between the daypart and TransactionNo variables to see what part of the day the traffic is
As we noted based on the result of this test, **the bakery may wish to consider increasing the number of staff during the noon and morning hours to handle the higher volume of transactions during those times**. In addition, the bakery may want to consider offering promotions or discounts during the less busy evening and night hours to attract more customers during those times.

Then we proceeded to the **third step in the exploration process, which is to perform statistical tests:**

We ran a statistical test to see if the conclusion they reached about the parts of the day that had transactions was just the result of chance first.

We tested by using chi-square steps for independence in SQL Server,
Where we concluded that we can reject the null hypothesis and conclude that there is a significant correlation between the periods of the day and the number of transactions.

Explore

iNterpret (N)

Based on the questions we need to answer, we conducted a market basket analysis to identify patterns and correlations among items purchased. Therefore, **the appropriate model to use at this point would be the association rule mining model.**

We used the **Python programming language** at this stage:

We first create association rules, which we will use to answer the following questions.

iNterpret (N)

Then we analyzed the results to answer the questions.
The **first question: What are the most frequently purchased items together?**

Analyzing the answer to this question, we concluded that “coffee” appears in 47.84% of all transactions, which means that it is the most frequently purchased item. The second row shows that "bread" appears in 32.72% of all transactions, making it the second most frequently purchased item. The third row shows that the word "Tea" appears in 14.26% of all transactions, making it the third most frequently purchased item.

We also concluded that the item combination (“Coffee”, “Bread”) accounted for 9.00% of all transactions, making it the most purchased two item combination. The next most purchased items are (“tea” and “cake”), (“coffee” and “cake”) and (“coffee” and “pastries”), appearing in 2.36%, 2.37% and 4.95% of all transactions, respectively.

Model

iNterpret (N)

The second question, how often do customers buy certain items together?

We extrapolate that customers who buy “cake” are likely to also buy “coffee”, with a support of 0.103856, which means that this group of items appears in 10.39% of all transactions.

- The confidence of this rule is 0.478394, which means that 47.84% of transactions containing “cake” also contain “coffee”. The base lift is 1.101515, which means that the observed support for this element group is 1.101515 times higher than what would be expected if “Cake” and “Coffee” were purchased independently. Other measures provide additional information about the strength and significance of the association.

iNterpret (N)

Question 3: Can we upsell or cross-sell certain items based on customer buying patterns?

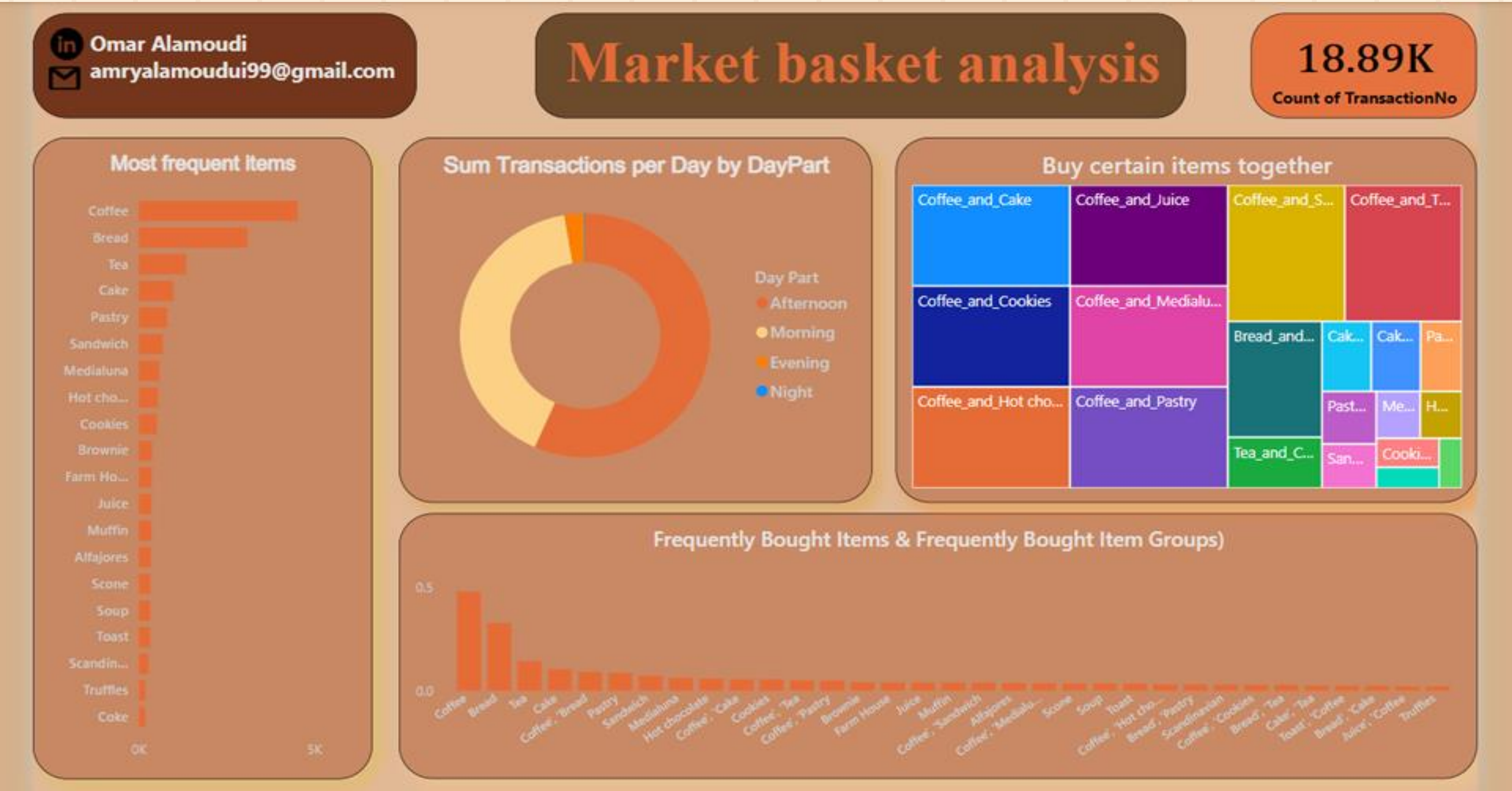
Market basket analysis of bakery data resulted in a list of association rules that define items that are most often bought together. Bases with higher leverage values indicate stronger correlations between elements.

For example, "tea" and "cake" are closely related, as are "coffee" and "toast". These associations can be leveraged to upsell items or sell them to customers who already buy one. Other rules, such as "hot chocolate" -> "coffee" and "bread" -> "pastries" can also be useful for promoting sales during certain seasons or times of the day.

Model

Dashboard

Using the Power Bi tool, we made a dashboard



Recommendations

Based on the results of the market basket analysis, stakeholders should consider the following recommendations:

1. Promote the purchase of “coffee” and “bread” together, as they are the most frequently purchased items and the most popular combination of the two.

2. Offer promotions or deals on frequently purchased items together, such as “coffee,” “cake,” “tea,” “cake,” “coffee,” and “pastries.”

3. Consider offering a variety of pastries to increase the popularity of the "coffee" and "pastry" combination.

4. Strategically placing 'coffee', 'bread' and 'tea' in high traffic areas to increase visibility and encourage purchase.

5. Encourage the purchase of "cake" and "coffee" together, as customers usually buy them together.

6. Underselling or upselling of items based on association rules generated by market basket analysis, such as promoting "hot chocolate" during the winter months.

By implementing these recommendations, stakeholders can improve sales and marketing strategies, enhance customer satisfaction, and improve product offerings, in accordance with project goals.

End