MARKET BASKET INSIGHTS

Project Title:

Market Basket Analysis

Phase-4:

Development part 2

Topic:

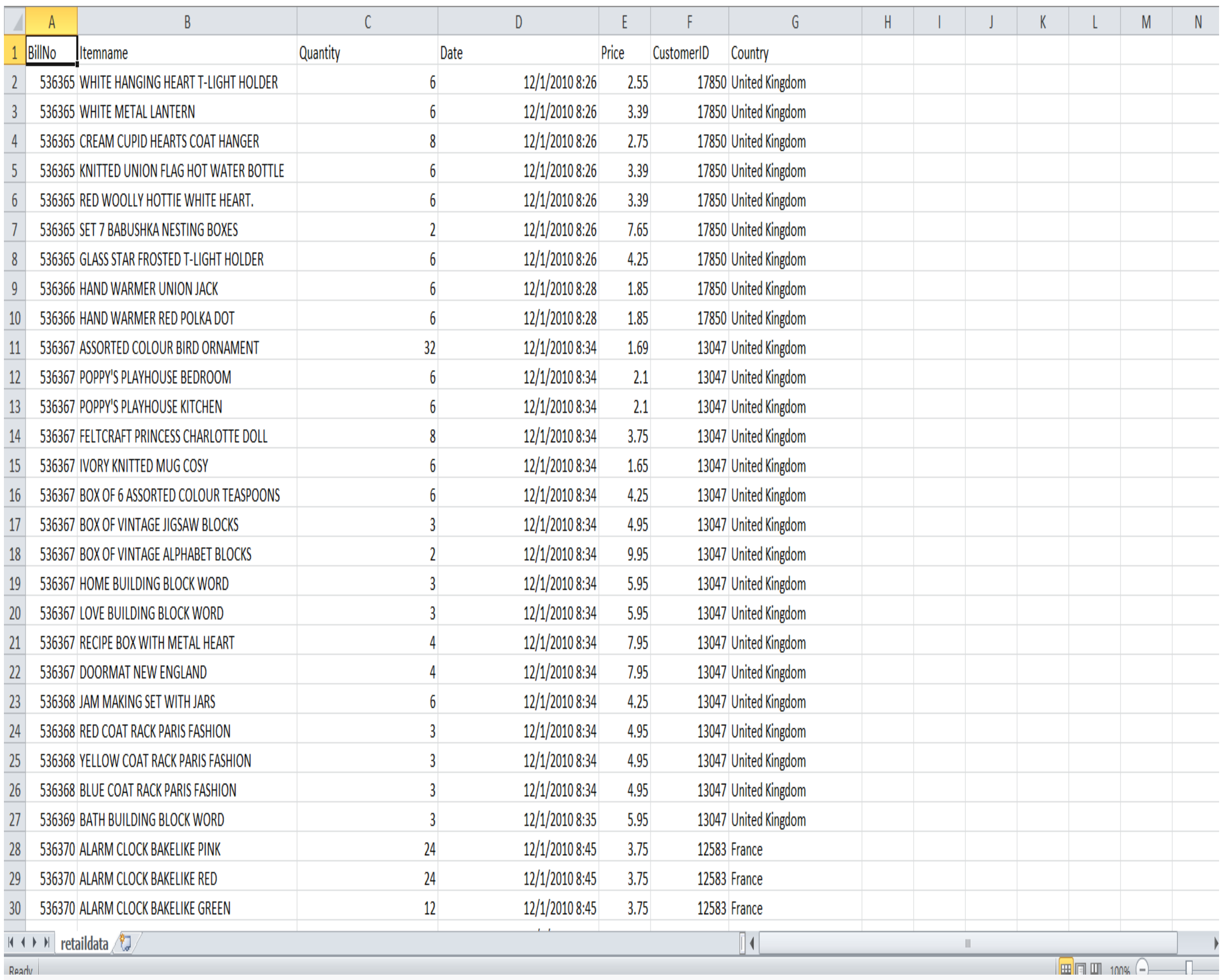
In this technology you will continue building your project by selecting a machine learining algorithm, training the model , and evaluating its performance. Perform different analysis as needed. After performing the relevant activities create a document around it and share the same for assessment.

Data Source:

A good data source for market basket analysis using analysis techniques

,Apriori algorithm to find frequently co-occuring products and generate insights for business optimization.

Dataset Link : (https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country							
2	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom							
3	536365	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom							
4	536365	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom							
5	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom							
6	536365	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom							
7	536365	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom							
8	536365	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom							
9	536366	HAND WARMER UNION JACK	6	12/1/2010 8:28	1.85	17850	United Kingdom							
10	536366	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1.85	17850	United Kingdom							
11	536367	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1.69	13047	United Kingdom							
12	536367	POPPY'S PLAYHOUSE BEDROOM	6	12/1/2010 8:34	2.1	13047	United Kingdom							
13	536367	POPPY'S PLAYHOUSE KITCHEN	6	12/1/2010 8:34	2.1	13047	United Kingdom							
14	536367	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	12/1/2010 8:34	3.75	13047	United Kingdom							
15	536367	IVORY KNITTED MUG COSY	6	12/1/2010 8:34	1.65	13047	United Kingdom							
16	536367	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	12/1/2010 8:34	4.25	13047	United Kingdom							
17	536367	BOX OF VINTAGE JIGSAW BLOCKS	3	12/1/2010 8:34	4.95	13047	United Kingdom							
18	536367	BOX OF VINTAGE ALPHABET BLOCKS	2	12/1/2010 8:34	9.95	13047	United Kingdom							
19	536367	HOME BUILDING BLOCK WORD	3	12/1/2010 8:34	5.95	13047	United Kingdom							
20	536367	LOVE BUILDING BLOCK WORD	3	12/1/2010 8:34	5.95	13047	United Kingdom							
21	536367	RECIPE BOX WITH METAL HEART	4	12/1/2010 8:34	7.95	13047	United Kingdom							
22	536367	DOORMAT NEW ENGLAND	4	12/1/2010 8:34	7.95	13047	United Kingdom							
23	536368	JAM MAKING SET WITH JARS	6	12/1/2010 8:34	4.25	13047	United Kingdom							
24	536368	RED COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom							
25	536368	YELLOW COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom							
26	536368	BLUE COAT RACK PARIS FASHION	3	12/1/2010 8:34	4.95	13047	United Kingdom							
27	536369	BATH BUILDING BLOCK WORD	3	12/1/2010 8:35	5.95	13047	United Kingdom							
28	536370	ALARM CLOCK BAKELIKE PINK	24	12/1/2010 8:45	3.75	12583	France							
29	536370	ALARM CLOCK BAKELIKE RED	24	12/1/2010 8:45	3.75	12583	France							
30	536370	ALARM CLOCK BAKELIKE GREEN	12	12/1/2010 8:45	3.75	12583	France							

retaildata

Ready 100%

Machine learning algorithms:

Market basket analysis is a common application of machine learning in retail and e-commerce to discover patterns and associations between items that are frequently purchased together.

The most popular algorithm for market basket analysis is the Apriori algorithm. However, there are other techniques and variations that can be used, depending on the specific requirements and size of your dataset. Here are some popular choices:

Apriori Algorithm:

Apriori is a classic algorithm for association rule mining, particularly for market basket analysis. It identifies frequent itemsets and generates association rules based on support and confidence levels.

FP-growth Algorithm:

The FP-growth (Frequent Pattern growth) algorithm is an alternative to Apriori that is more efficient in terms of memory and runtime. It builds a compact data structure called an FP-tree to mine frequent itemsets.

Eclat Algorithm:

Eclat (Equivalence Class Transformation) is another algorithm for frequent itemset mining. It uses a depth-first search approach and is known for its simplicity and efficiency.

FPGrowth Algorithm:

FPGrowth (Frequent Pattern Growth) is

a variation of FP-growth that works well with large datasets and is implemented in libraries like Spark's MLlib.

Training the model:

Training a machine learning model, regardless of the specific algorithm you choose, involves several key steps. Here is a high-level overview of the typical process for training a machine learning model:

Data Collection: Gather and prepare a dataset that includes historical or training data. This data should consist of input features (attributes) and corresponding output labels or target values that the model needs to learn to predict.

Data Preprocessing: Clean and preprocess the data to ensure it is in a suitable format for training. This may involve tasks such as handling missing values, encoding categorical variables, scaling features, and splitting the data into training and testing sets.

Feature Engineering: Depending on the specific problem and dataset, you may need to engineer or create new features that can improve the model's ability to learn patterns and make predictions effectively.

Choosing a Model: Select the machine learning algorithm or model architecture that is appropriate for your problem. The choice of model depends on factors like the type of data, the nature of the problem (classification, regression, clustering, etc.), and your specific goals.

Model Training: Train the selected model using the training data. During training, the model learns to make predictions by adjusting its internal parameters to minimize a predefined loss or error function. This involves iterations or epochs, and the model gradually improves its performance.

<u>**Python program:**</u>

```
from sklearn.model_selection

import train_test_split

from sklearn.tree import
DecisionTreeClassifier

from sklearn.metrics import
accuracy_score

X = [[5.1, 3.5, 1.4, 0.2],

    [4.9, 3.0, 1.4, 0.2],
     [6.3, 3.3, 6.0, 2.5],
     # ... more data ...
   ]
y = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2]  # Target labels
(0, 1, 2, ...)


X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy}")
```

Evaluate the performance of the algorithm:

The performance of the Apriori algorithm in market basket analysis can vary based on several factors, including the size of the dataset, the hardware and software used, and the specific parameters and implementation of the algorithm. Here are some considerations regarding the performance of the Apriori algorithm in market basket insights:

Scalability: Apriori can be computationally expensive, especially when dealing with large transaction datasets with many items. The algorithm has to generate a large number of candidate itemsets, and this process can become slow as the dataset size increases.

Thresholds: The performance of Apriori is influenced by the minimum support and confidence thresholds you set. Lower support thresholds can result in more frequent itemsets but may increase computational complexity. Finding the right balance is essential.

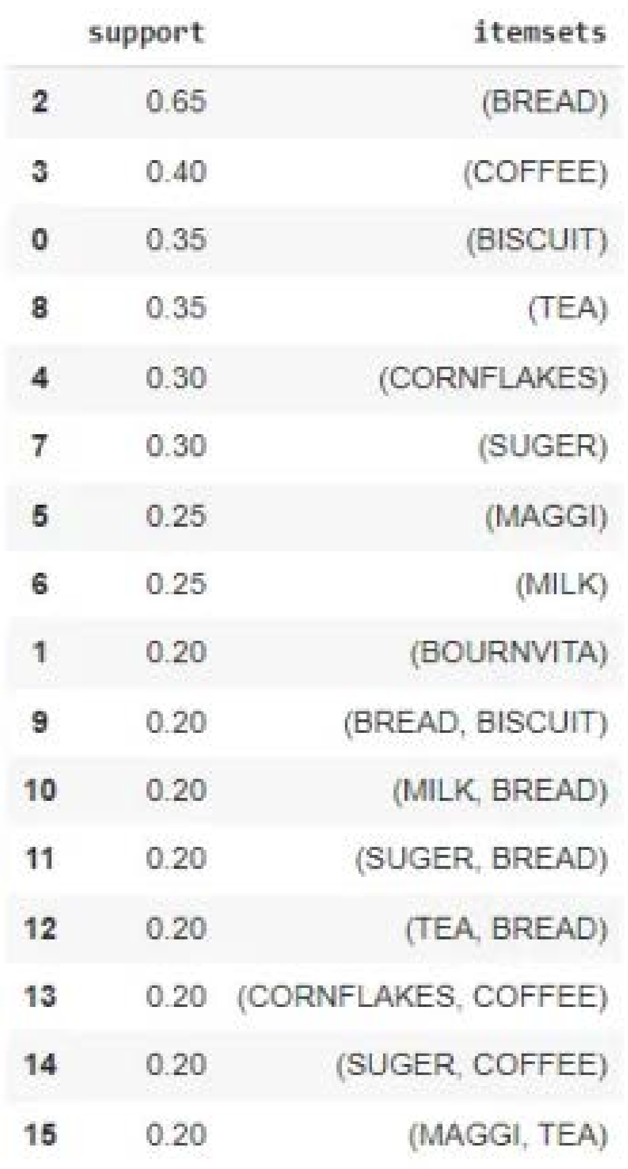
Data Preprocessing: Data preprocessing, such as reducing the number of unique items or filtering out infrequent items, can significantly impact the algorithm's performance. Cleaning the data and removing noise is important.

Algorithm Optimization: There are various optimization techniques and variations of the Apriori algorithm that can improve its performance, such as the use of hash-based techniques and pruning strategies. Choosing an optimized implementation can

Source Code

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
df = pd.read_csv("Assignment-1_Data.csv", names = ['transaction'], sep = ',')
df = list(df["Itemname].apply(lambda x:x.split(",")))
one_hot_transformer = TransactionEncoder()
df_transform = one_hot_transformer.fit_transform(df)
df = pd.DataFrame(df_transform,columns=one_hot_transformer.columns_)
df = apriori(df, min_support = 0.2, use_colnames = True)
df.sort_values(['support'],ascending=False, inplace = True)
df_ar = association_rules(df, metric="lift", min_threshold=1)
```

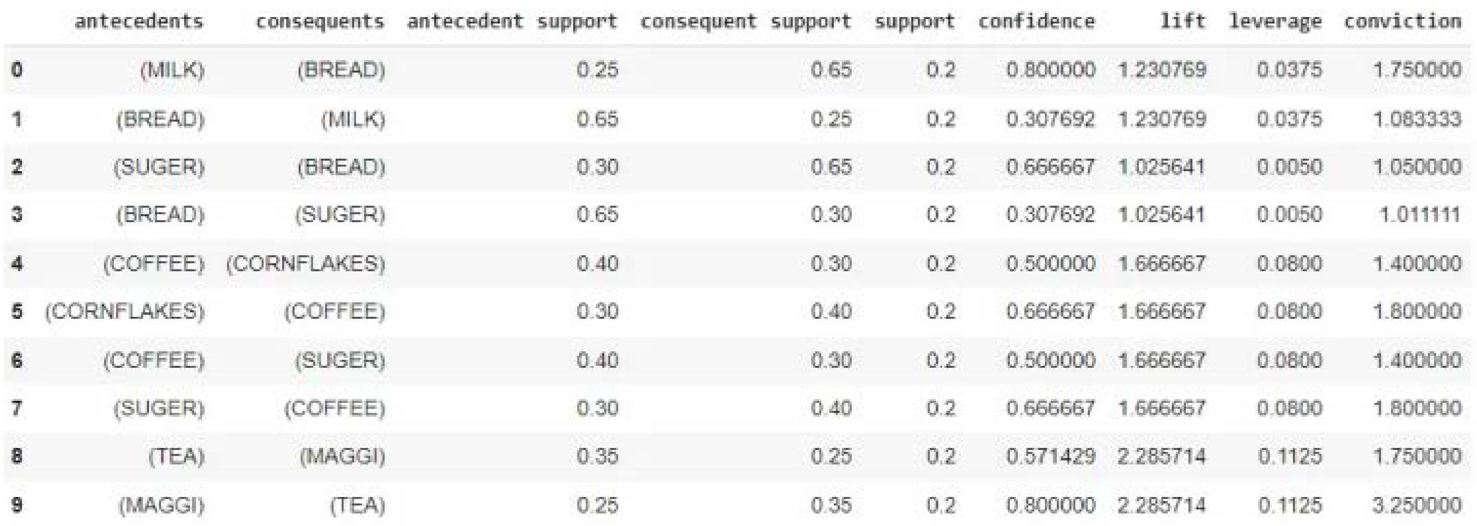
Find the frequent itemsets using Apriori



	support	itemsets
2	0.65	(BREAD)
3	0.40	(COFFEE)
0	0.35	(BISCUIT)
8	0.35	(TEA)
4	0.30	(CORNFLAKES)
7	0.30	(SUGER)
5	0.25	(MAGGI)
6	0.25	(MILK)
1	0.20	(BOURNVITA)
9	0.20	(BREAD, BISCUIT)
10	0.20	(MILK, BREAD)
11	0.20	(SUGER, BREAD)
12	0.20	(TEA, BREAD)
13	0.20	(CORNFLAKES, COFFEE)
14	0.20	(SUGER, COFFEE)
15	0.20	(MAGGI, TEA)

Association rule

The association_rules function will automatically calculate key metrics of our transaction data including support, confidence, lift, leverage, and conviction.



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	0.0375	1.750000
1	(BREAD)	(MILK)	0.65	0.25	0.2	0.307692	1.230769	0.0375	1.083333
2	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	0.0050	1.050000
3	(BREAD)	(SUGER)	0.65	0.30	0.2	0.307692	1.025641	0.0050	1.011111
4	(COFFEE)	(CORNFLAKES)	0.40	0.30	0.2	0.500000	1.666667	0.0800	1.400000
5	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.800000
6	(COFFEE)	(SUGER)	0.40	0.30	0.2	0.500000	1.666667	0.0800	1.400000
7	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	0.0800	1.800000
8	(TEA)	(MAGGI)	0.35	0.25	0.2	0.571429	2.285714	0.1125	1.750000
9	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	0.1125	3.250000