Online-Retail-Recommendation-Analysis

Concepts Used:-

- PCA
- K-Means Clustering
- Agglomerative - Hierarchical clustering
- Apriori Algorithm

Abstract:

The project is based on the customer segmentation of the online retail dataset. The aim of our model is to give analytical reports based on different parameters of a particular customer's buying behavior. Throughout the model, we have also done a comparative analysis of two unsupervised algorithms:- KMeans and agglomerative - hierarchical clustering.
In addition to this, we have tried to implement a new model that is specifically dedicated to customer behavior datasets, known as the Apriori model.

Introduction:

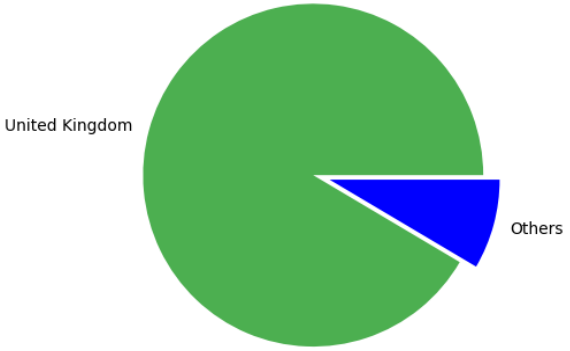
In these days of higher demand for retail products, it has become necessary for shop-owners to analyze customer behavior, so as to maximize profit and customer satisfaction. Thus, customer segmentation becomes necessary. Customer segmentation is the process of dividing a group of people into groups with similar buying characteristics. Clustering, a class of unsupervised machine learning algorithms, can accomplish this. We have followed the steps such as ideation of the model - pipeline, preprocessing of data and visualization, segmentation of data using various clustering algorithms mentioned above, and finally, visualization and interpretation of results.

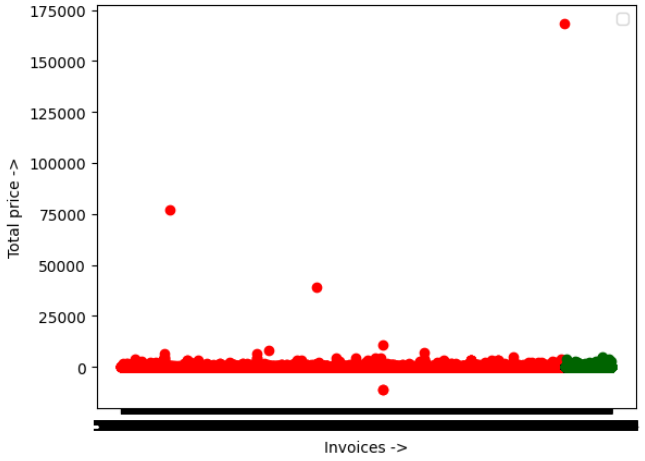
Methodology:

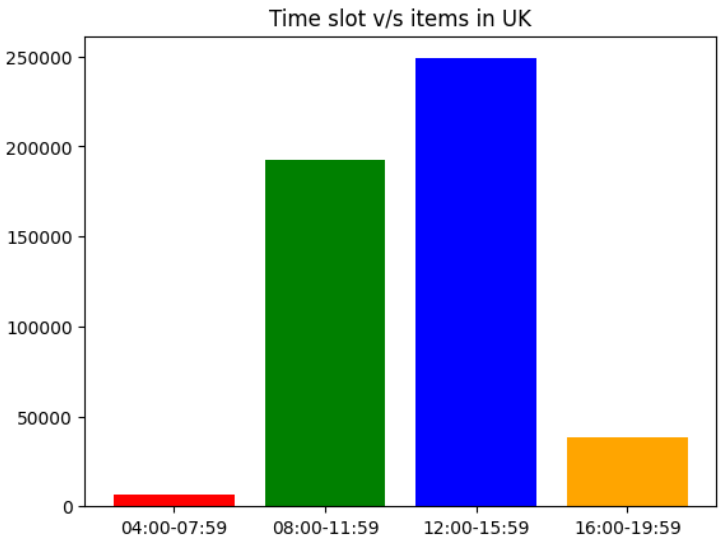
Preprocessing and Visualization:

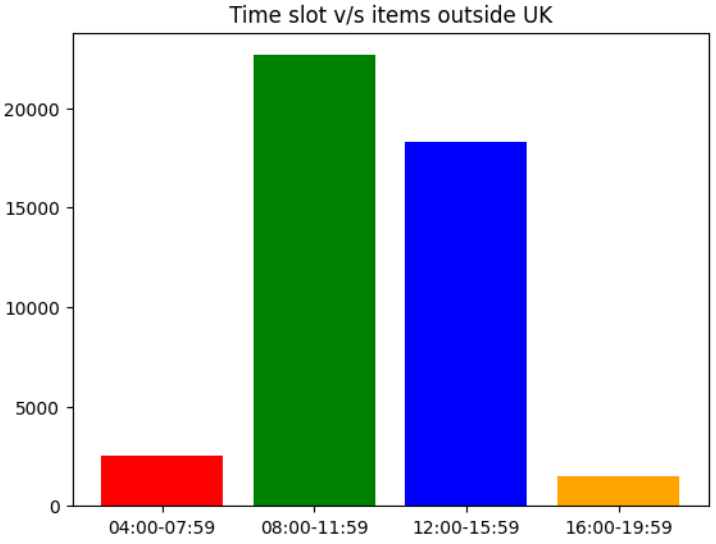
The project is started off with pre-processing of the online retail csv. We proceed to extract those features which will actually be useful for clustering . From the 'InvoiceDate' column, we have separated the dates,months,s and times ; Of which months and times are useful. Further, we have made time intervals from 4:00 - 7:59, 8:00 - 11:59, 12:00 - 15:59 , 16:00 - 19:59 . A column indicating the months has been created.

Following that, a relevant 'TotalPrice" = Quantity*Unit_Price column was created. For grouping on the basis of countries, we observe that the no. of customers from the United Kingdom are almost 480,000 million, and the rest of the countries are 45,000. Thus, we divide them into 'United Kingdom' and 'Not United Kingdom'.
The visualization of data is shown as :-









Functions used during the analysis:

1. **person_df (df):**

   This function takes customer-behavior-segmented dataframe from the input and outputs a dataframe with features relevant for clustering.These are features such as mean purchases,sum of purchases, item count , total quantity , avg quantity . It also has CustomerID, which is not relevant for clustering , but is used to finally track items bought, from the original dataframe , for recommendations.

2. **df_cluster_labels(df,k) :**
   Applies KMeans on the input dataset(which is obtained from person_df) with the input values of k. *Note that it does not take into account the feature 'CustomerID' for clustering*. It returns the input dataset , with 'cluster_labels' as a new column. This column will help us in visualizing the clusters properly.
3. **descript_dict(og_df, person_df):**
   Takes the final 'person_df' and extracts the item descriptions using the CustomerIDs .It returns a k number of dataframe with the customerID and corresponding item descriptions , where k is the number of clusters made.
4. **plot_dendrogram(model,**kwargs):**
   Creates a linkage matrix and then plots the dendrogram showing the hierarchy of the clusters obtained by the hierarchical clustering algorithm. Also creates the counts of samples under each node

5. **Agglocls(df,cluster_count):**
   Runs an agglomerative clustering algorithm over the input dataframe df and clusters the datapoints in a hierarchical manner. It takes the dataframe and number of clusters as input and returns a dictionary. The dictionary has cluster number as label and a numpy array of corresponding datapoints stored in it.

6. **AggloDendo(df):**
   Trains the agglomerative model on the input dataset and then calls the plot_dendogram function to build the dendrogram.

7. **AggloPCA(df):**
   Trains the PCA agglomerative model on the input dataset and then calls the plot_dendogram function to build the dendrogram.

8. **processlis(lis):**
   Returns a new list which has no. of samples greater than or equal to 100. This function is used to apply apriori algorithm on data frames of samples with size greater than 100.

Procedure in brief -

We have used the month and time-based lists' dataframes. Such as:-

	InvoiceNo	StockCode	Description	Quantity	UnitPrice	CustomerID	Country	time_intervals	Total Price	Money_bins
209029	555156	23299	FOOD COVER WITH BEADS SET 2	6	3.75	2422	United Kingdom	04:00-07:59	22.50	0
209030	555156	22847	BREAD BIN DINER STYLE IVORY	1	16.95	2422	United Kingdom	04:00-07:59	16.95	0
209031	555157	23075	PARLOUR CERAMIC WALL HOOK	16	4.15	2422	United Kingdom	04:00-07:59	66.40	0
209032	555157	47590B	PINK HAPPY BIRTHDAY BUNTING	6	5.45	2422	United Kingdom	04:00-07:59	32.70	0
209033	555157	22423	REGENCY CAKESTAND 3 TIER	4	12.75	2422	United Kingdom	04:00-07:59	51.00	0
...	...	...	...	...	...	...	...	...	...	...
245898	558637	22032	BOTANICAL LILY GREETING CARD	12	0.42	4051	United Kingdom	16:00-19:59	5.04	0
245899	558637	22028	PENNY FARTHING BIRTHDAY CARD	12	0.42	4051	United Kingdom	16:00-19:59	5.04	0
245900	558637	22033	BOTANICAL ROSE GREETING CARD	12	0.42	4051	United Kingdom	16:00-19:59	5.04	0
245901	558637	22029	SPACEBOY BIRTHDAY CARD	12	0.42	4051	United Kingdom	16:00-19:59	5.04	0
245902	558637	22024	RAINY LADIES BIRTHDAY CARD	12	0.42	4051	United Kingdom	16:00-19:59	5.04	0

36056 rows × 10 columns

At a time, one of this dataframe is acted upon by person_df(.) to give a completely new data frame containing new relevant features.This new data represents each customer as per their buying habits, on which we can cluster to group people with similar purchase habits.

After applying KMeans clustering , we append the cluster labels.

	No. of visits	Mean_purch	Sum_purch	Item_count	Total_quantity	Avg_quant	CustomerID
0	1	21.251111	382.52	18	196	10.888889	1
1	1	23.602553	1109.32	47	356	7.574468	12
2	1	22.992000	459.84	20	211	10.550000	28
3	1	16.876389	607.55	36	293	8.138889	29
4	1	19.890000	99.45	5	20	4.000000	31
...	...	...	...	...	...	...	...
987	1	21.440000	214.40	10	92	9.200000	4318
988	1	18.570870	427.13	23	177	7.695652	4320
989	1	11.545714	80.82	7	54	7.714286	4335
990	2	2.770541	307.53	77	209	1.882883	4337
991	183	11.332004	100526.21	1904	30638	3.453726	4339

992 rows × 7 columns

Before Clustering

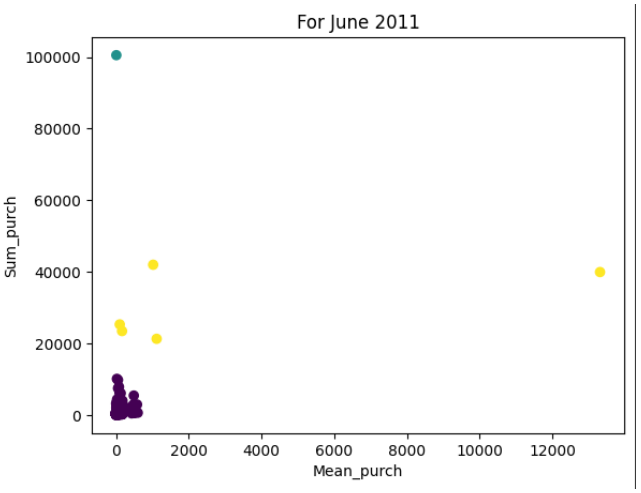
	No. of visits	Mean_purch	Sum_purch	Item_count	Total_quantity	Avg_quant	CustomerID	cluster_labels
0	1	21.251111	382.52	18	196	10.888889	1	0
1	1	23.602553	1109.32	47	356	7.574468	12	0
2	1	22.992000	459.84	20	211	10.550000	28	0
3	1	16.876389	607.55	36	293	8.138889	29	0
4	1	19.890000	99.45	5	20	4.000000	31	0
...	...	...	...	...	...	...	...	...
987	1	21.440000	214.40	10	92	9.200000	4318	0
988	1	18.570870	427.13	23	177	7.695652	4320	0
989	1	11.545714	80.82	7	54	7.714286	4335	0
990	2	2.770541	307.53	77	209	1.882883	4337	0
991	183	11.332004	100526.21	1904	30638	3.453726	4339	1

992 rows × 8 columns

After Clustering

However, for most of these conditional(In a specific month or time or country bin) dataframes we have outlier clusters , which have very few data points in some clusters as well, such as:-

Name: cluster_labels
0 986
2 5
1 1



This is backed by the fact that it has a very high silhouette score = 0.9421. Generally, a silhouette score close to 1 is assumed to be very good, however, here we can see the outliers, thus this high value of silhouette score is unrealistic.
These clusters can also be visualized as shown in the adjoining figure.

Thus, to remove outliers, we remove data points with outlier cluster labels and we re run the KMeans pipeline as :-

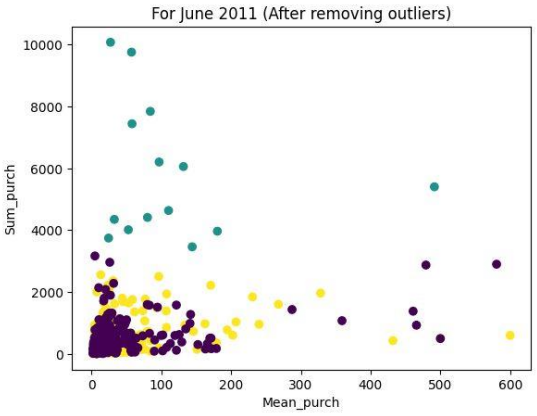
Removed cluster labels:-

	No. of visits	Mean_purch	Sum_purch	Item_count	Total_quantity	Avg_quant	CustomerID
0	1	21.251111	382.52	18	196	10.888889	1
1	1	23.602553	1109.32	47	356	7.574468	12
2	1	22.992000	459.84	20	211	10.550000	28
3	1	16.876389	607.55	36	293	8.138889	29
4	1	19.890000	99.45	5	20	4.000000	31
...	...	...	...	...	...	...	...
986	1	23.300000	69.90	3	18	6.000000	4316
987	1	21.440000	214.40	10	92	9.200000	4318
988	1	18.570870	427.13	23	177	7.695652	4320
989	1	11.545714	80.82	7	54	7.714286	4335
990	2	2.770541	307.53	77	209	1.882883	4337

New cluster labels are as

	No. of visits	Mean_purch	Sum_purch	Item_count	Total_quantity	Avg_quant	CustomerID	cluster_labels
0	1	21.251111	382.52	18	196	10.888889	1	1
1	1	23.602553	1109.32	47	356	7.574468	12	1
2	1	22.992000	459.84	20	211	10.550000	28	1
3	1	16.876389	607.55	36	293	8.138889	29	1
4	1	19.890000	99.45	5	20	4.000000	31	1
...	...	...	...	...	...	...	...	...
986	1	23.300000	69.90	3	18	6.000000	4316	2
987	1	21.440000	214.40	10	92	9.200000	4318	2
988	1	18.570870	427.13	23	177	7.695652	4320	2
989	1	11.545714	80.82	7	54	7.714286	4335	2
990	2	2.770541	307.53	77	209	1.882883	4337	2

The value counts of the cluster labels are:-
Name: cluster_labels
1 497
2 475
0 14
This clustering comes out to be relatively good. The new value of silhouette score is now realistic, i.e silhouette score = 0.5340 . This implies that the clustering is done relatively good in this case.



From this properly clustered dataframe, we obtained various different dataframes, according to the customerID and the description of the corresponding items that they bought. It is shown as :-

```
[      CustomerID                                        Description
0             1  [RABBIT NIGHT LIGHT, REGENCY TEA STRAINER, REG...
1            12  [JUMBO  BAG BAROQUE BLACK WHITE, SET OF 4 PANT...
2            28  [RED RETROSPOT MINI CASES, RED RETROSPOT PURSE...
3            29  [LUNCH BAG APPLE DESIGN, CHILDRENS CUTLERY SPA...
4            31  [WHITE WOOD GARDEN PLANT LADDER, TRAY, BREAKFA...
..          ...                                                ...
492        2166  [SET OF 3 REGENCY CAKE TINS, DOILEY STORAGE TI...
493        2168  [SMALL POPCORN HOLDER, JUMBO BAG VINTAGE LEAF,...
494        2169                                          [Manual]
495        2171  [CHILDRENS CUTLERY RETROSPOT RED , RED RETROSP...
496        2173  [REGENCY SUGAR BOWL GREEN, SET OF 3 REGENCY CA...

[497 rows x 2 columns],
     CustomerID                                        Description
0            50  [COLOUR GLASS T-LIGHT HOLDER HANGING, WHITE ME...
1            70  [LARGE SKULL WINDMILL, PINK HAPPY BIRTHDAY BUN...
2           562  [ANTIQUE SILVER TEA GLASS ENGRAVED, LED TEA LI...
3           997  [CLASSIC FRENCH STYLE BASKET NATURAL, CLASSIC ...
4          1285  [ALARM CLOCK BAKELIKE IVORY, ALARM CLOCK BAKEL...
5          1334  [REGENCY CAKESTAND 3 TIER, PETIT TRAY CHIC, WH...
6          1435  [SET 6 PAPER TABLE LANTERN STARS , RABBIT NIGH...
7          1880  [STRAWBERRY RAFFIA FOOD COVER, PACK OF 20 NAPK...
8          2177  [JUMBO BAG ALPHABET, PAPER BUNTING RETROSPOT, ...
9          2571  [JUMBO BAG STRAWBERRY, JUMBO BAG RED RETROSPOT...
10         2703  [FRYING PAN UNION FLAG, JUMBO BAG PINK POLKADO...
11         2991  [PACK OF 6 SKULL PAPER PLATES, PACK OF 20 SKUL...
12         3729  [SPOTTY BUNTING, BAKING MOULD CHOCOLATE CUPCAK...
13         3772  [SET 6 PAPER TABLE LANTERN HEARTS , SET OF 60 ...,
     CustomerID                                        Description
0          2191  [STRAWBERRY CERAMIC TRINKET BOX, REGENCY CAKES...
1          2195  [HANGING HEART JAR T-LIGHT HOLDER, VICTORIAN G...
2          2206  [HEART OF WICKER LARGE, WHITE WIRE EGG HOLDER,...
3          2211  [RED RETROSPOT CHARLOTTE BAG, CHARLOTTE BAG PI...
4          2212  [BALLOON WATER BOMB PACK OF 35, CARD CAT AND T...
..          ...                                                ...
470        4316  [PARTY BUNTING, CLASSICAL ROSE TABLE LAMP, FRE...
471        4318  [VINTAGE  2 METER FOLDING RULER, LUNCH BAG APP...
472        4320  [PARISIENNE JEWELLERY DRAWER , PARISIENNE CURI...
473        4335  [ROBOT BIRTHDAY CARD, CARD CIRCUS PARADE, PENN...
474        4337  [WHITE HANGING HEART T-LIGHT HOLDER, DOILEY ST...
```

Finally, the Apriori algorithm is applied on these dataframes to obtain the rules for customer recommendations.
The Apriori algorithm is a data mining algorithm that identifies frequent sets of items that occur together in transactions or events. It uses a bottom-up approach to generate frequent itemsets and prune infrequent ones.This pruning is done as per the support value which is user-specified and the support value is inversely proportional to the time complexity of this algorithm.

In this case, the support values for all the datasets are set to 0.075 because the normal computers don't have enough computational power for lower support values.
The Apriori algorithm is used here, to generate association rules between the items in the frequent itemsets, which can be used for prediction and pattern recognition.
Thus, the frequent itemsets obtained after applying on the KMeans-obtained dataset is :-

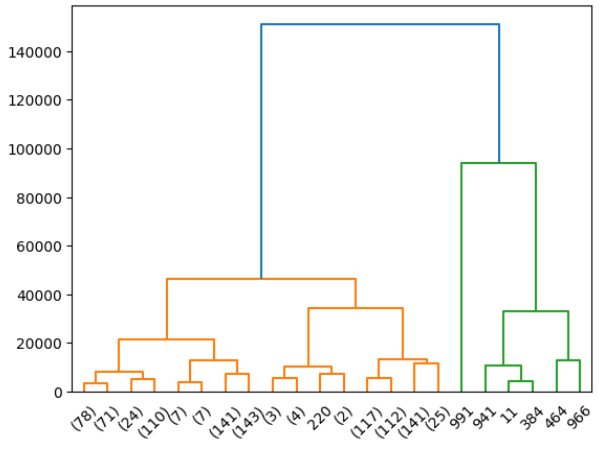
```
Frequent Itemsets:
     support                                           itemsets
0       0.08                        (ALARM CLOCK BAKELIKE PINK)
1       0.08                        (ASSORTED COLOURS SILK FAN)
2       0.10                     (CHARLOTTE BAG APPLES DESIGN)
3       0.09                   (CHILDRENS CUTLERY DOLLY GIRL )
4       0.09                     (CHILDRENS CUTLERY SPACEBOY )
..       ...                                              ...
73      0.08      (SET/20 RED RETROSPOT PAPER NAPKINS , POSTAGE)
74      0.09                      (POSTAGE, SPACEBOY LUNCH BOX )
75      0.11  (ROUND SNACK BOXES SET OF4 WOODLAND , ROUND SN...
76      0.08  (PLASTERS IN TIN WOODLAND ANIMALS, POSTAGE, PL...
77      0.11  (ROUND SNACK BOXES SET OF4 WOODLAND , POSTAGE,...
```

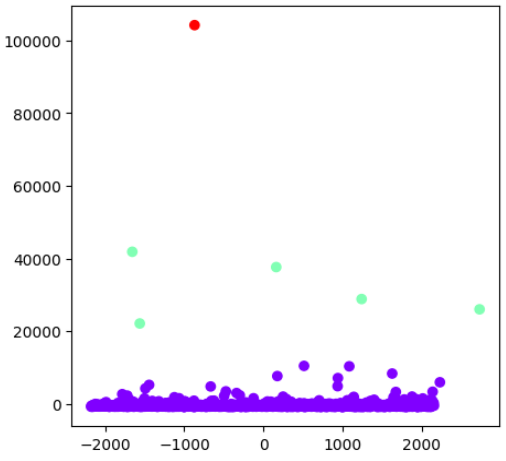
Frequent itemsets are the frequently occurring items in the given dataset. The methods of utilizing association_rules and frequent_itemsets is mentioned in the analysis section.
The association rules are as follows:-

Association Rules:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(CHARLOTTE BAG APPLES DESIGN)	(POSTAGE)	0.10	0.71	0.09	0.900000	1.267606	0.0190	2.900000
1	(POSTAGE)	(CHARLOTTE BAG APPLES DESIGN)	0.71	0.10	0.09	0.126761	1.267606	0.0190	1.030645
2	(POSTAGE)	(CHILDRENS CUTLERY DOLLY GIRL)	0.71	0.09	0.09	0.126761	1.408451	0.0261	1.042097
3	(CHILDRENS CUTLERY DOLLY GIRL)	(POSTAGE)	0.09	0.71	0.09	1.000000	1.408451	0.0261	inf
4	(CHILDRENS CUTLERY SPACEBOY)	(POSTAGE)	0.09	0.71	0.08	0.888889	1.251956	0.0161	2.610000
..	...	...	...	...	...	...	...	...	...
57	(ROUND SNACK BOXES SET OF4 WOODLAND , ROUND SN...	(POSTAGE)	0.11	0.71	0.11	1.000000	1.408451	0.0319	inf
58	(POSTAGE, ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.16	0.11	0.11	0.687500	3.819444	0.0812	2.624000
59	(ROUND SNACK BOXES SET OF4 WOODLAND)	(POSTAGE, ROUND SNACK BOXES SET OF 4 FRUITS)	0.18	0.16	0.11	0.611111	3.819444	0.0812	2.160000
60	(POSTAGE)	(ROUND SNACK BOXES SET OF4 WOODLAND , ROUND SN...	0.71	0.11	0.11	0.154930	1.408451	0.0319	1.053167
61	(ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND , POSTAGE)	0.16	0.17	0.11	0.687500	4.044118	0.0828	2.656000

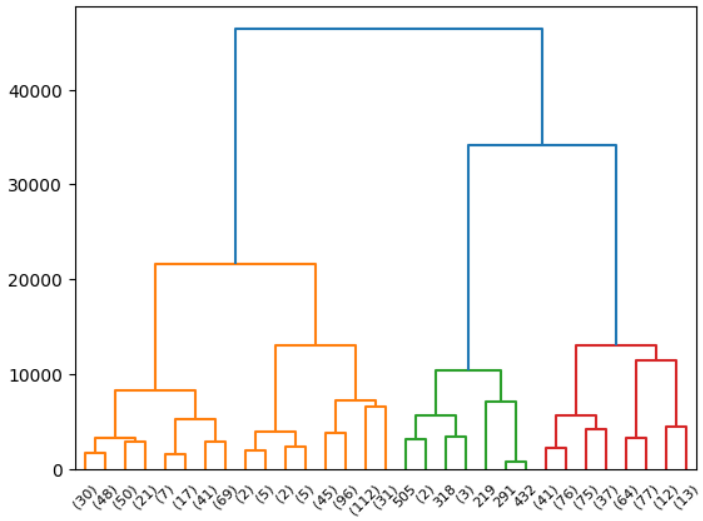
Now, coming to the hierarchical method of clustering. Firstly, the data frame is transformed using PCA (n_components = 3). The dendrograms for the hierarchical agglomerative algorithm is as follows:-



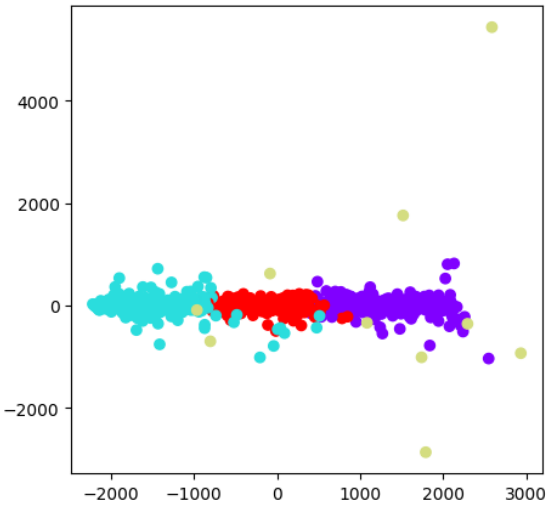


Which indicates the same issue as seen in K-Means processing. Thus, the outliers are removed and we see:-

Dendrogram:-



New clusters(More refined):-



After applying get_descrip(.) function, we obtain the customerID and corresponding items' list and we apply Apriori algorithm on it to get association_rules and frequent_itemsets as in KMeans and for agglomerative clustering . The above analysis is performed for all the data frames separated by months and time.
All of this data is stored inside a dictionary called 'Rules_dict' .

Analysis:

Frequent_itemsets:

Retailer can analyze the frequent itemsets in the following ways:-

- Stock Optimization: The retailer can identify which items are bought frequently , and accordingly avoid overstocking and understocking of inventory and meet the customer demands.
- Store Layout: Store layout can be optimized as per the frequent selling itemsets.
- Pricing optimization: By analyzing the frequent_itemsets, the retailer can set prices of the frequent purchase products based on customer demand and competitive pricing in the market.

Association_rules:

Understanding a few metrics:-

1. Support: Proportion of transactions in the dataset that contain both items in an association rule.High support value means itemsets are common and are good candidates for association mining.

2. Confidence: Confidence is the proportion of transactions that contain the antecedent of the association rule that also contain the consequent.
3. Conviction: Conviction measures how much the antecedent of the association rule is dependent on the consequent.A high conviction value indicates that the rule has a strong predictive power and is likely to be useful for making recommendations.

Retailer can analyze the frequent itemsets in the following ways:-

- Product bundling: The retailer can use the frequent itemsets to identify products that are often purchased together and create product bundles or package deals. This can help them increase sales and revenue by encouraging customers to purchase multiple products at once.
- Customer Segmentation: The association rules can be used to segment customers based on their purchasing behavior. For example, the retailer can identify customers who frequently purchase certain products and target them with a promotion package that includes all the items bought together by the customer.
- Product recommendations: The frequent itemsets can be used to make personalized product recommendations to customers. For example, if a customer purchases product A, the retailer can suggest other products frequently purchased with product A to the customer.
- Promotions and discounts: The frequent itemsets can help the retailer identify products that are good candidates for promotions or discounts. For example, if a particular product is frequently purchased with another product, the retailer can offer a discount on the second product to incentivize customers to make the purchase.

Preprocessing:-

Preprocessing, along with the association rules aids in demand forecasting. Demand forecasting is the process of estimating future demand for a product or service. In retail, demand forecasting is used to predict how many products will be sold in the future based on past sales, trends, and other factors.

Advantages and Drawbacks:

- **Advantages:**
    - The method analyzes the data for month wise and timewise basis which helps in overall demand and inventory analysis based on various types of clusters for that particular month or time rather than a general clustering done for the entire year
    - By splitting month wise we can account for festivals and other special occasions as well while understanding consumer behavior. Hence any special items which are in demand for a specific time period only are better analyzed compared to an overall clustering which might downplay a few of these occasional but significant items.

- **Drawbacks :**
    - Recommendation models like apriori and fp-growth models are very expensive in terms of computation. Because of this we have to reduce the datasets to 100-150 data points for each cluster. For larger datasets the programme crashed on both google collab and local machines.
    - The data is uneven in the dataset ie. less than 10 individuals have shopped multiple times in the month of october. Similar observations for time slots 12:00 - 16:00. Hence the model trained here gives out very biased recommendations.

Conclusion:

In conclusion, our study applied clustering and Apriori model machine learning techniques to analyze an online retail dataset. We did a comparative study on the K-Means clustering algorithm and Hierarchical clustering(agglomerative model) along with the Apriori model, which showed common buying patterns and links between items.

These insights can inform online retailers on how to improve their recommendations and tailor their marketing strategies to better meet customer needs.However, the limitations of this study include the very high sample size, its computation size and potential bias of the data. By utilizing the power of machine learning algorithms such as clustering and Apriori, online retailers can gain deeper insights into customer behavior, improve their product recommendations, and ultimately enhance their overall business performance.