Task2: To predict other two products in a transactions that have highest chance of buying when we buy some item

Important: I've explained most things in my jupyter notebook regarding analysis and the explanation of code cells

I approached this problem as a **Market Basket Analysis problem** in which we use **apriori algorithm** which comes under **association analysis** in particular. So, I first cleaned the data and the major discrepancies in it were that the **ItemDescription feature was not present in 2908 cases**, so I removed those NaN values. Since **there were some negative CostPerItems and NumberOfItemsPurchased**, so I removed them as well. Also I checked whether the **ItemCode of any item is negative, found out there were 12** of them out of which 9 were of no use because they were mostly for services and fees, **but the other 3 ("BOYS PARTY BAG", "GIRLS PARTY BAG", "PADS TO MATCH ALL CUSHIONS") were important so I changed their ItemCode to 1, 2, 3 respectively.** I also changed the format of TransactionTime to DateTime format which is of no use in this task albeit I did it. I did typecasting ItemDescription and TransactionId to string and removed the TransactionId which contained 'C' in it because that meant it is for the credit account which is of no use to us. Then I made a **Basket Dataframe** in **which for each transaction reported the number of specific items that were bought where items represented columns and TransactionId as rows**. Then I formed a basket_sets dataframe in which I did hot encoding i.e if value>0, I inputted 1 and when value=0, I inputted 0. Now for the **association analysis** I used Apriori algorithm in which I defined my rules, obtained frequent_itemsets and **set the metric upon which calculations will be done to be 'Lift'**(explained in Code with a snippet). I sorted the rules in descending order of lift values to get the prioritised elements first when they are clubbed in respect to antecedents. Showed the variation of **Suppor**t(It's the default popularity of an item. In mathematical terms, the support of item A is nothing but the ratio of transactions involving A to the total number of transactions.) with **Confidence**(Likelihood that customer who bought both A and B. It divides the number of transactions involving both A and B by the number of transactions involving B.) Also plotted the variation of **support with lift**. Now I made a **count dataframe** in which I stored how much of each item is bought by the user and the total items bought by them. I plotted the **histogram plot of the total items purchased by them** and observed that the data is **vastly skewed**, **meaning, within a certain range/number of items, there are a lot of customer that are present and mostly it is on the lower side(low in value) which makes sense that most people don't spend alot on different type of product when they buy or transact just once.** I tried to observe this trend keeping the **total_bought items < 25000** and then it actually showed the trend which was **a peak in the low range and exponential decrease afterwards.** I also plotted to see the trend on **log scale and the result was the plot was somewhat gaussian in nature**. Next I observed which item is bought the most time which was having Item code: **1787079('JUMBO BAG')** and how much every country bought the items, **the most being the UK followed by Netherlands**. With all this in hand, **I started the recommendation task in which I first converted the type of each of the antecedents and consequents from frozenset to list to make manipulations easier.I got the couplets of ItemCodes in descending order of their lift values with first one being the antecedent and second being consequent and put that in list 'l'. Made a function find_next_recommendation which would return the next best recommendation of the inputted item on a priority basis of lift values. Finally, I run that function for each of the antecedents present to get the consequents outputs on a priority basis.**

Lastly I got the csv output file named **'Output.csv'**