Q1： Association rules with Apriori

1. Filter out the count attribute as this will not be included in the rule generation.

I use the .drop() method in pd.dataframe to delete this column.

2. Use the Apriori algorithm to generate frequent itemsets from the input data. When doing so, only select frequent itemsets with a support of at least 15% (so, the minimum support should be 0.15).

There are 19 frequent items produced. 13 of them is length 1 and 7 of them is length 2.

3. Save the generated itemsets in ./output/question1 out apriori.csv,making sure to include the support column.

4. Using these frequent itemsets, generate a first batch of association rules with a minimum confidence of 0.9. How many rules are produced? For each rule, include a short description in your report.

1 rule were produced.

antecedents	consequents	antecedent	consequent	support	confidence	lift	leverage	conviction
frozenset({'21...25'})	frozenset({'junior'})	0.16	0.44	0.16	1	2.2727273	0.0896	inf

It described a rule from {‘21...25’} to {‘junior’}, the support is from the min of antecedent support and consequent support. The confidence is higher than the minimum confidence which is 0.9. Because it is a perfect confidence score, the denominator of the conviction becomes 0 (due to 1 - 1). So it is defined as 'inf'.

5. Generate a second batch of association rules, but this time use a minimum confidence of 0.7. How many rules are produced this time? Again, shortly describe the outcome in your report.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
frozenset({'21...25'})	frozenset({'junior'})	0.16	0.44	0.16	1	2.2727273	0.0896	inf
frozenset({'Ph.D'})	frozenset({'26...30'})	0.2	0.32	0.16	0.8	2.5	0.096	3.4
frozenset({'philosophy'})	frozenset({'26...30'})	0.28	0.32	0.2	0.7142857	2.2321429	0.1104	2.38

Three rules were produced this time: a rule from {‘21...25’} to {‘junior’}, a rule from {‘Ph.D’} to {‘26...30’}, a rule from{‘philosophy’} to {‘26...30’}.

Q2： Association rules with FP-Growth

1. Filter out the id attribute as this will not be include in the rule generation.

I use the .drop() method in pd.dataframe to delete this column.

2. Discretize the numeric attributes into 3 bins of equal width, the filter out the original attributes. When doing so, only select frequent itemsets with a support of at least 20% (so, the minimum support should be 0.2).

I use .cut() method in pandas to generate the bins and make the original attribute equal to the new attributes.

I also use pd.get_dummies to generate a new pandas SparseDataFrame for later use. Lastly I use fpgrowth in mlxtend.frequent_patterns to select frequent itemsets with a support of at least 20%.

3. Use the FP-Growth algorithm to generate frequent itemsets from the data. How many frequent itemsets are produced? How big are they? Include this information in your report.

230 frequent itemsets are produced. The length of them includes 1,2,3,4.

4. Using the obtained frequent itemsets, generate association rules. Experiment with different confidence values, selecting a value that produces at least 10 rules. What is this value? Include it in your report.

I set the min_threshold=0.78 , and 16 rules were produced.

5. 5.Select the top 2 most interesting rules and for each specify the following in your report:

1.

frozenset({'car_NO', 'mortgage_NO'})	frozenset({'current_act_YES'})

anteceden	consequen	support	confidenc	lift	leverage	conviction
0.3283333	0.7583333	0.2633333	0.8020305	1.0576226	0.0143472	1.2207265

2.

frozenset({'save_act_YES', 'region_INNER_CITY'})	frozenset({'current_act_YES'})

anteceden	consequen	support	confidenc	lift	leverage	conviction
0.2883333	0.7583333	0.2266667	0.7861272	1.0366512	0.0080139	1.129955

I find these two rules interesting because their first attribute is relatively simple which makes it easier to find connection and second attribute is current_act__yes that is useful and they have relatively high confidence and high conviction which is close to perfect rule.