COMP40370 Practical 3

Ryan Jennings 19205824

Question 1: Association rules with Apriori

1) To start the lab the dataset is read in. The count column is not needed for the calculations so it is deleted
   `del df_1['count']`
2) The second part of question 1 is where the Apriori algorithm comes in. The `mxltend` python library is imported in order to use the apriori method and also the association rules method for the rest of question 1.
   First the data is formatted from a dataframe into an array of arrays:
   ```
   records = []
   for i in range(0, len(df_1)):
   records.append([str(df_1.values[i, j]) for j in range(0, len(df_1.columns))])
   ```
   The records must be encoded first before using the apriori method. We can use mxltend's example in their documentation for this:
   ```
   te = TransactionEncoder()
   te_ary = te.fit(records).transform(records)
   df_ap = pd.DataFrame(te_ary, columns=te.columns_)
   ```
   Then it is fed into the apriori method with a minimum support of 15%:
   `ap = apriori(df_ap, min_support=0.15, use_colnames=True)`
   This produced 20 frequent itemsets. 13 of which contained only one item and 7 that contained 2 items.
3) Part 1.3 just involves writing a dataframe to a csv
   `ap.to_csv('./output/question1_out_apriori.csv', index=False, float_format='%g')`
4) The next section makes use of another `mxltend` library method, this time to generate association rules from frequent itemsets, association_rules. The function is simply called with a 90% confidence threshold
   ```
   assoc_rules_90_pc = association_rules(ap, metric="confidence", min_threshold=0.9)
   ```
   This returns just one association rule that is both over 15% support and 90% confidence that 16% of the dataset guarantees that students added between 21 and 25 will be a junior.
5) Again we write out the dataset using the same method with a different file name as 1.3
6) We repeat 1.4 but with a minimum threshold of 70%. With this we would expect to see more records. We do see more records up from 1 record to now 3 records. With confidence values of 0.8 and 0.71.
   That 100% of students aged between 21 and 25 are juniors.
   That 80% of PhD students will be between 26 and 30 years old and
   that 71% of Philosophy students will be between 26 and 30 as well.

7) The results of our association rules are written to a file just as for 1.5.

Question 2: Data Reduction and Discretisation

1) The Id attribute is filtered out just the same way as question 1.1
2) The attributes age, income and children were discretized with the pandas cut method for creating equal width bins.
3) The FP-Growth produces 231 frequent itemsets with the largest containing 3 attributes.
4) The frequent itemsets are written to a csv with the pandas to_csv method
5) I found that a value of 0.79 produced exactly 10 association rules. With the lowest of this set having a confidence score of 0.7905
6) The association rules are written to a csv with the pandas to_csv method
7) Two of the most interesting rules in my opinion were:
    a) Only 79% of people aged between 50 and 67 have savings accounts. This could be that they are not counting a pension as a savings account. If not then this is a worrying figure and shows 21% of older people might struggle without work. The bank could start a marketing campaign giving 1000 EUR to older people that open a savings account in their bank. This would drive up use of their bank and also give older people a source for financial stability.
    b) 84% of people who are married have a current account but no mortgage or pep. This could be a sign of the times that people just cannot find houses for a mortgage. Outside of the bank's obvious lack of ability to build houses of which we are in desperate need of supply, the banks could lower their interest rates on mortgages and add benefits and discounts for first time buyers of new or used houses.