HW 2 Part 2

Q1a)

Names of the attributes: ['Unnamed: 0', 'CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT', 'Price']

Q1b)

'RAD' and 'TAX' are highly correlated with a correlation coefficient of 0.910. The correlation coefficient is positive, thus meaning there is a strong positive linear relationship between these two features.

Q2)

Min sup = 0.001%
Min conf = 2%

```
# Applying Apriori to find frequent itemsets with a minimum support
frequent_itemsets = apriori(df, min_support=0.00001, use_colnames=True)

# Generating association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.02)

# Sort rules by lift
#rules = rules.sort_values(['confidence', 'lift'], ascending=[False, False])

rules = rules.sort_values(['lift'], ascending=[False])

# Display the filtered rules
rules[['antecedents', 'consequents', 'antecedent support', 'consequent support', 'support', 'confidence'
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
40	(perfume, bed_bath_table)	(market_place)	0.000010	0.002879	0.00001	1.000000	347.342857
42	(telephony, cool_stuff)	(cine_photo)	0.000062	0.000668	0.00001	0.166667	249.374359
24	(auto, fashion_bags_accessories)	(musical_instruments)	0.000010	0.006457	0.00001	1.000000	154.866242
31	(garden_tools, bed_bath_table)	(construction_tools_lights)	0.000041	0.002509	0.00001	0.250000	99.647541
14	(fashion_sport)	(fashion_female_clothing)	0.000278	0.000401	0.00001	0.037037	92.360874

Min sup = 0.0001%
Min conf = 25%

```
# Applying Apriori to find frequent itemsets with a minimum support
frequent_itemsets = apriori(df, min_support=0.000001, use_colnames=True)

# Generating association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.25)

# Sort rules by lift
#rules = rules.sort_values(['confidence', 'lift'], ascending=[False, False])

rules = rules.sort_values(['lift'], ascending=[False])

# Display the filtered rules
rules[['antecedents', 'consequents', 'antecedent support', 'consequent support', 'support', 'confidence'
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
12	(perfume, bed_bath_table)	(market_place)	0.000010	0.002879	0.00001	1.00	347.342857
4	(auto, fashion_bags_accessories)	(musical_instruments)	0.000010	0.006457	0.00001	1.00	154.866242
6	(garden_tools, bed_bath_table)	(construction_tools_lights)	0.000041	0.002509	0.00001	0.25	99.647541
3	(auto, musical_instruments)	(fashion_bags_accessories)	0.000010	0.019166	0.00001	1.00	52.175966
8	(bed_bath_table, construction_tools_lights)	(garden_tools)	0.000010	0.036173	0.00001	1.00	27.645253

Min sup = 0.001%
Min conf = 50%

```
# Applying Apriori to find frequent itemsets with a minimum support
frequent_itemsets = apriori(df, min_support=0.00001, use_colnames=True)

# Generating association rules
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)

# Sort rules by lift
#rules = rules.sort_values(['confidence', 'lift'], ascending=[False, False])

rules = rules.sort_values(['lift'], ascending=[False])

# Display the filtered rules
rules[['antecedents', 'consequents', 'antecedent support', 'consequent support', 'support', 'confidence'
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
8	(perfume, bed_bath_table)	(market_place)	0.00001	0.002879	0.00001	1.0	347.342857
2	(auto, fashion_bags_accessories)	(musical_instruments)	0.00001	0.006457	0.00001	1.0	154.866242
1	(auto, musical_instruments)	(fashion_bags_accessories)	0.00001	0.019166	0.00001	1.0	52.175966
5	(bed_bath_table, construction_tools_lights)	(garden_tools)	0.00001	0.036173	0.00001	1.0	27.645253
9	(telephony, cine_photo)	(cool_stuff)	0.00001	0.037345	0.00001	1.0	26.777533