Question 1

The following contingency table summarizes the survey data of a student population, where ***bike*** refers to students who bike, refers to students who do not bike, ***ski*** refers to students who ski, and refers to students who do not ski.

	ski	***Not ski***	***Sum(row)***
bike	600	700	1300
Not bike	1900	800	2700
Sum(col)	2500	1500	4000

(a) Based on the given data, determine the correlation relationship between bike and ski using the *lift* measure. (5 pts)

(b) Suppose that the association rule “bike ⇒ *ski* ” is mined. Given a minimum support threshold of 15% and a minimum confidence threshold of 40%, is this association rule strong (i.e., meet the thresholds)? (5 pts)

Your Answer:

a. Lift = Support(X,Y)/(Support(X)*Support(Y)) = 0.15/(0.325*0.625) = 0.738

b. Support(bike -> ski) = 600/4000 = 0.15

Confidence(bike -> ski) = 600/1300 = 0.462

Yes, the association rule is strong by these criteria.

1a) Nature of correlation between bike and ski is not concluded. (-1)

Question 2

Given a data set with five transactions, each containing five items, as shown in the table.

TID	***items_bought***
T1	{A, K, T, X, Z}
T2	{A, H, X, T, Z}
T3	{A, B, D, R, S}
T4	{B, D, H, T, X}

T5	{B, C, H, M, S}

(a) What is the maximum number of possible frequent itemsets? (5 pts)

(b) Let *min_support* = 40%. Find all frequent itemsets using the Apriori algorithm. Your answer should include the key steps of the computation process. (5 pts)

(c) In the computation (b) above, how many rounds of database scan are needed? What is the total number of candidates? (5 pts)

(d) Let *n* be the total number of transactions, *b* be the number of items in each transaction, *m* be the number of *k*-itemset candidates. Consider the following two different approaches for counting the support values of the candidates. For each transaction, the first approach checks if a candidate occurred in the transaction or not; the second approach enumerates all the possible *k*-itemsets of the transaction and checks if the itemset is one of the candidates. What is the computation complexity for each approach? Is one always better than the other? (**Optional, 5-point extra credit**)

Your Answer:

a. Since there are 12 unique items and five items per transaction, the maximum possible number of frequent item sets is Combination(12,1) + Combination(12,2) + Combination(12,3) + Combination(12,4) + Combination(12,5) = 12 + 66 + 220 + 495 + 792 = 1573.

b. Item sets of size one that meet the minimum support threshold are {A}, {B}, {D}, {T}, {X}, {Z}, {H}, and {S}. This means that only item sets including two of these items are to be checked. There are Combination(8,2) = 28 of these candidates.

Of these possible item sets of size 2, the ones that meet the minimum support threshold are {A,X}, {A,T}, {A,Z}, {T,X}, {T,Z}, {Z,X}, {H,X}, {H,T}, {B,S}, {B,D}, and {B,H}. Therefore, the only size 3 item sets we need to check are ones where each of the contained item sets of size 2 meet the threshold, which are {A,T,X}, {A,T,Z}, {A,X,Z}, {T,X,Z}, and {H,X,T}.

All of these item sets of size 3 meet the minimum support threshold, so the size 4 item sets we need to check are ones where each of the contained item sets of size 3 meet the threshold, which is only {A,X,T,Z}

{A,X,T,Z} meets the minimum support threshold. No item sets of size 5 need to be checked.

c. The database is scanned 4 times, once to find the support of all item sets of size 1 then once for k = 2, 3, and 4 once candidate item sets have been generated. There are 12 candidates for size one item sets, 28 for size 2, 5 for size 3 and 1 for size 4, so 46 total candidates.

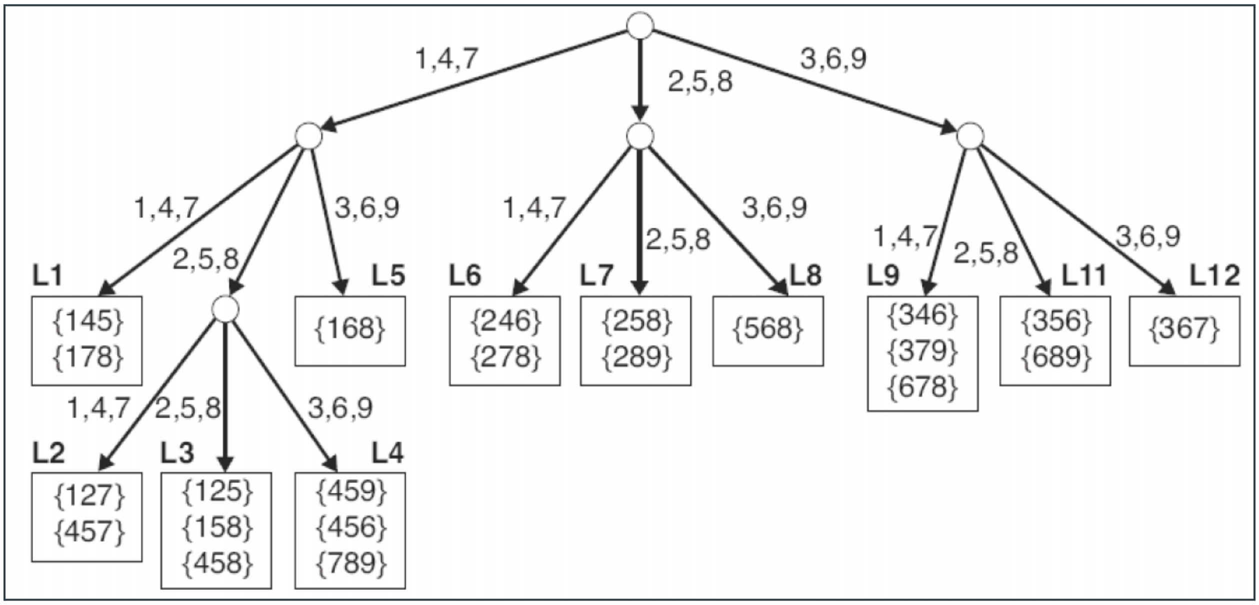
Question 3

In the Apriori algorithm, we can use a hash tree data structure to efficiently count the support of candidate itemsets. Consider the hash tree for candidate 3-itemsets shown in the figure below.

(a) Based on this figure, how many candidate 3-itemsets are there in total? (5 pts)

(b) Given a transaction that contains items {1, 2, 5, 6, 9}, which of the hash tree leaf nodes will be visited when finding the candidate 3-itemsets contained in the transaction? (5 pts)

(c) Use the visited leaf nodes in part (b) to determine the candidate itemsets that are contained in the transaction {1, 2, 5, 6, 9}. (5 pts)



Your Answer:

a. There are a total of 22 candidate itemsets of size 3.

b. The possible item sets of size 3 for this transaction are {125}, {126}, {129}, {156}, {159}, {169}, {256}, {259}, {269}, and {569}. Traversing to try to find these item sets results in visiting the leaf nodes L3, L4, L5, L7, and L8.

c. The only candidate item set from the tree that is in this transaction is {125}.

Question 4

Given a data set with four transactions. Let *min_support* = 70%, and *min_confidence* = 80%.

customer_ID	***TID***	***items_bought* (in the form of brand-item category)**
01	T100	{Farmer's-Milk, Wonder-Bread, Sweet-Pie, Sunny-Cherry}
02	T200	{Dairyland-Cheese, Farmer's-Milk, Goldenfarm-Cherry, Sweet-Pie, Wonder-Bread}
01	T300	{King’s-Cereal, Sunset-Milk, Dairyland-Cheese, Best-Bread}
03	T400	{Wonder-Bread, Farmer's-Milk, Best-Cereal, Sweet-Pie, Dairyland-Cheese}

(a) At the granularity of *item_category* (e.g., $item_i$ could be “*Milk*” and ignore brand name), for the following rule template,

$$\forall X \in \textbf{transaction}, \mathit{buys}(X, item_1) \wedge \mathit{buys}(X, item_2)) \Rightarrow \mathit{buys}(X, item_3)\ [s, c]$$

list the frequent *k*-itemset(s) for the largest *k*, and all of the strong association rules (with their support *s* and confidence *c*) containing the frequent *k*-itemset(s) for the largest *k*. Your answer should include the key steps of the computation process. (10 pts)

(b) At the granularity of *brand−item_category* (e.g., item$_i$ could be “King’s-Cereal”), for the following rule template,

$$\forall X \in \textbf{customer},\ \ buys(X, item_1) \land buys(X, item_2)) \Rightarrow buys(X, item_3)$$

list the frequent k-itemset(s) for the largest k (but do not print any rules). Your answer should include the key steps of the computation process. (10 pts)

Your Answer:

a. Frequent item sets at k = 1: {Bread}, {Milk}, {Cheese}, {Pie}

Candidate item sets for k = 2: {Bread, Milk}, {Bread, Cheese}, {Bread, Pie}, {Cheese, Milk}, {Milk, Pie}, {Cheese, Pie}

Frequent item sets at k = 2: {Bread, Milk}, {Bread, Cheese}, {Bread, Pie}, {Cheese, Milk}, {Milk, Pie}

Candidate item sets for k = 3: {Bread, Milk, Cheese}, {Bread, Milk, Pie}

Frequent item sets for k = 3: {Bread, Milk, Cheese}, {Bread, Milk, Pie}

No candidate item sets for k = 4

Candidate association rules based on frequent item sets of k = 3: (Bread, Milk) -> Cheese, (Bread, Milk) -> Pie, (Bread, Cheese) -> Milk, (Cheese Milk) -> Bread, (Bread, Pie) -> Milk, (Pie, Milk) -> Bread

Association rules based on frequent item sets of k = 3 that meet the minimum confidence: (Bread, Cheese) -> Milk, (Cheese, Milk) -> Bread, (Bread, Pie) -> Milk, (Pie, Milk) -> Bread

b.

Frequent item sets at k = 1: {Farmer's-Milk}, {Wonder-Bread}, {Sweet-Pie}, {Dairyland-Cheese}

Candidate item sets for k = 2: {Farmer's-Milk, Wonder-Bread}, {Farmer's-Milk, Dairyland-Cheese}, {Farmer's-Milk, Sweet-Pie}, {Dairyland-Cheese, Wonder-Bread}, {Wonder-Bread, Sweet-Pie}, {Dairyland-Cheese, Sweet-Pie}

Frequent item sets at k = 2: {Farmer's-Milk, Wonder-Bread}, {Farmer's-Milk, Dairyland-Cheese}, {Farmer's-Milk, Sweet-Pie}, {Dairyland-Cheese, Wonder-Bread}, {Wonder-Bread, Sweet-Pie}, {Dairyland-Cheese, Sweet-Pie}

Candidate item sets for k = 3: {Farmer's-Milk, Wonder-Bread, Dairyland-Cheese}, {Farmer's-Milk, Wonder-Bread, Sweet-Pie}, {Farmer's-Milk, Sweet-Pie, Dairyland-Cheese}, {Sweet-Pie, Wonder-Bread, Dairyland-Cheese}

Frequent item sets for k = 3: {Farmer's-Milk, Wonder-Bread, Dairyland-Cheese}, {Farmer's-Milk, Wonder-Bread, Sweet-Pie}, {Farmer's-Milk, Sweet-Pie, Dairyland-Cheese}, {Sweet-Pie, Wonder-Bread, Dairyland-Cheese}

Candidate item sets for k = 4: {Farmer's-Milk, Wonder-Bread, Dairyland-Cheese, Sweet-Pie}

Frequent item sets for k = 4: {Farmer's-Milk, Wonder-Bread, Dairyland-Cheese, Sweet-Pie}

Candidate association rules based on frequent item sets of k = 4: (Wonder-Bread, Farmer's-Milk) -> Dairyland-Cheese, (Wonder-Bread, Farmer's-Milk) -> Sweet-Pie, (Wonder-Bread, Dairyland-Cheese) -> Farmer's-Milk, (Dairyland-Cheese, Farmer's-Milk) -> Wonder-Bread, (Wonder-Bread, Sweet-Pie) -> Farmer's-Milk, (Sweet-Pie, Farmer's-Milk) -> Wonder-Bread, (Dairyland-Cheese, Sweet-Pie) -> Wonder-Bread, (Dairyland-Cheese, Sweet-Pie) -> Farmer's-Milk, (Dairyland-Cheese, Farmer's-Milk) -> Sweet-Pie, (Dairyland-Cheese, Wonder-Bread) -> Sweet-Pie, (Sweet-Pie, Farmer's-Milk) -> Dairyland-Cheese, (Sweet-Pie, Wonder-Bread) -> Dairyland-Cheese

Association rules based on frequent item sets of k = 4 that meet the minimum confidence: (Wonder-Bread, Farmer's-Milk) -> Dairyland-Cheese, (Wonder-Bread, Farmer's-Milk) -> Sweet-Pie, (Wonder-Bread, Dairyland-Cheese) -> Farmer's-Milk, (Dairyland-Cheese, Farmer's-Milk) -> Wonder-Bread, (Wonder-Bread, Sweet-Pie) -> Farmer's-Milk, (Sweet-Pie, Farmer's-Milk) -> Wonder-Bread, (Dairyland-Cheese, Sweet-Pie) -> Wonder-Bread, (Dairyland-Cheese, Sweet-Pie) -> Farmer's-Milk, (Dairyland-Cheese, Farmer's-Milk) -> Sweet-Pie, (Dairyland-Cheese, Wonder-Bread) -> Sweet-Pie, (Sweet-Pie, Farmer's-Milk) -> Dairyland-Cheese, (Sweet-Pie, Wonder-Bread) -> Dairyland-Cheese

Question 5

Consider the heart disease data set shown in the following table.

Diabetes	**High Blood Pressure**	**Smoking**	**Exercise**	**Heart Disease**
Yes	No	Non-smoker	Yes	No
No	No	Occasional smoker	Yes	No
Yes	No	Occasional smoker	Yes	No
No	Yes	Former smoker	No	Yes
Yes	No	Frequent smoker	No	Yes
No	Yes	Occasional smoker	Yes	No
No	Yes	Former smoker	Yes	Yes
Yes	Yes	Non-smoker	Yes	Yes
No	No	Frequent smoker	Yes	Yes
No	Yes	Non-smoker	No	Yes
Yes	No	Frequent smoker	No	Yes
No	No	Former smoker	Yes	Yes

Let *Heart Disease* be the class label. **Show the key steps for the following tasks**.

(a) Using information gain as the attribute selection measure, construct the first level of the decision tree. (15 pts)

(b) If gain ratio is used as the attribute selection measure, will the first level of the decision tree be different from above? (5 pts)

(c) Given someone with the following attribute values: *Diabetes* = “No”, *High Blood Pressure* =“No”, *Smoking* = “Non-smoker”, and *Exercise* = “Yes”, how would a naïve Bayesian classifier determine whether *Heart Disease* would be Yes or No? Show your computation. (15 pts)

Your Answer:

a. Info(D) = I(8,4) = -2/3*log_2(2/3) -1/3*log_2(1/3) = 0.918

Info_diabetes(D) = 5/12*I(3,2) + 7/12*I(5,2) = 5/12*(-3/5*log_2(3/5) -2/5*log_2(2/5)) + 7/12*(-5/7*log_2(5/7) -2/7*log_2(2/7)) = 0.908

Info_highbp(D) = 5/12*I(4,1) + 7/12*I(4,3) = 5/12*(-4/5*log_2(4/5) -1/5*log_2(1/5)) + 7/12*(-4/7*log_2(4/7) -3/7*log_2(3/7)) = 0.876

Info_smoker(D) = 1/4*I(2,1) + 1/4*I(3,0) + 1/4*I(3,0) + 1/4*I(0,3) = 1/4*(-2/3*log_2(2/3) -1/3*log_2(1/3)) + 0 + 0 + 0 = 0.230

Info_exercise(D) = 2/3*I(4,4) + 1/3*I(4,0) = 2/3*(-1/2*log_2(1/2) -1/2*log_2(1/2)) + 0 = 0.667

The best variable to split on is "smoker." within this variable because its information gain (Info(D) - Info_smoker(D)) is the highest of all predictor variables. It's best to split "occasional smoker" instances onto one side and classify it as "no" and all other instances onto the other side and classify it as "yes" because this maximized the classification accuracy at 11/12. From there, the occasional smoker node should be a leaf node because it has no imuurity while the other node should be split further because it still has impurity.

b. This split will not change if gain ratio is used instead of information gain is used because the "smoker" variable also has the highest information gain (0.918/0.230).

c. P(Diabetes = no) = 7/12

P(Highbp = no) = 7/12

P(Smoker = non smoker) = 1/4

P(Exercise = yes) = 2/3

P(Diabetes = no | Heart disease = yes) = 5/8

P(Highbp = no | Heart disease = yes) = 1/2

P(Smoker = non smoker | Heart disease = yes) = 1/4

P(Exercise = yes | Heart disease = yes) = 1/2

P(Heart disease) = 2/3

Thus the probability of heart disease according to a Naive Bayes classifier based on this data is ((2/3) * (5/8) * (1/2) * (1/4) * (1/2))/((7/12) * (7/12) * (1/4) * (2/3)) = 0.459