Q1:

(1) tabular table is below:

Cuboid	Location	Time	Item	SUM(Quantity)
LTI	Sydney	2005	PS2	1400
LTI	Sydney	2006	PS2	1500
LTI	Sydney	2006	Wii	500
LTI	Melbourne	2005	XBox 360	1700
LT	Sydney	2005	ALL	1400
LT	Sydney	2006	ALL	2000
LT	Melbourne	2005	ALL	1700
LI	Sydney	ALL	PS2	2900
LI	Sydney	ALL	Wii	500
LI	Melbourne	ALL	XBox 360	1700
TI	ALL	2005	PS2	1400
TI	ALL	2006	PS2	1500
TI	ALL	2006	Wii	500
TI	ALL	2005	XBox 360	1700
L	Sydney	ALL	ALL	3400
L	Melbourne	ALL	ALL	1700
T	ALL	2005	ALL	3100
T	ALL	2006	ALL	2000
I	ALL	ALL	PS2	2900
I	ALL	ALL	Wii	500
I	ALL	ALL	XBox 360	1700
	ALL	ALL	ALL	5100

(2) the SQL statement is:

```
SELECT Location, Time, Item, SUM(Quantity)
FROM Sales
GROUP BY Location, Time, Item
UNION ALL
SELECT Location, Time, ALL, SUM(Quantity)
FROM Sales
GROUP BY Location, Time
UNION ALL
SELECT Location, ALL, Item, SUM(Quantity)
FROM Sales
GROUP BY Location, Item
UNION ALL
```

```
SELECT ALL, Time, Item, SUM(Quantity)
FROM Sales
GROUP BY Time, Item
UNION ALL
SELECT Location, ALL, ALL, SUM(Quantity)
FROM Sales
GROUP BY Location
UNION ALL
SELECT ALL, Time, ALL, SUM(Quantity)
FROM Sales
GROUP BY Time
UNION ALL
SELECT ALL, ALL, Item, SUM(Quantity)
FROM Sales
GROUP BY Item
UNION ALL
SELECT ALL, ALL, ALL, SUM(Quantity)
FROM Sales
```

(3) after query, the tabular table is:

Cuboid	Location	Time	Item	SUM(Quantity)
LT	Sydney	2006	ALL	2000
LI	Sydney	ALL	PS2	2900
L	Sydney	ALL	ALL	3400
T	ALL	2005	ALL	3100
T	ALL	2006	ALL	2000
I	ALL	ALL	PS2	2900
	ALL	ALL	ALL	5100

(4) The second function $f(x) = 16 \cdot fLocation(x) + 4 \cdot fTime(x) + fItem(x)$ is feasible, the reason is blew:

Draw the MOLAP cube of two functions:

Cuboid	Location	Time	Item	SUM(Quantity)	$f(x) = 9 \cdot fLocation(x) + 3 \cdot fTime(x) + fItem(x)$	$f(x) = 16 \cdot fLocation(x) + 4 \cdot fTime(x) + fItem(x)$
LTI	1	1	1	1400	13	21
LTI	1	2	1	1500	16	25
LTI	1	2	3	500	18	27
LTI	2	1	2	1700	23	38
LT	1	1	0	1400	12	20

LT	1	2	0	2000	15	24
LT	2	1	0	1700	21	36
LI	1	0	1	2900	10	17
LI	1	0	3	500	12	19
LI	2	0	2	1700	20	34
TI	0	1	1	1400	4	5
TI	0	2	1	1500	7	9
TI	0	2	3	500	9	11
TI	0	1	2	1700	5	6
L	1	0	0	3400	9	16
L	2	0	0	1700	18	32
T	0	1	0	3100	3	4
T	0	2	0	2000	6	8
I	0	0	1	2900	1	1
I	0	0	3	500	3	3
I	0	0	2	1700	2	2
	0	0	0	5100	0	0

We can see from the above table, for f(x) = 9 · fLocation(x) + 3 · fT ime(x) + fItem(x), it has some same index eg:3, 9,12, so, it is not the one to one function, for mapping, it needs the one to one function. For f(x) = 16 · fLocation(x) + 4 · fTime(x) + fItem(x), the the index are unique, its each index is correspond to each value and no same index, so, the second function is feasible which is f(x) = 16 · fLocation(x) + 4 · fTime(x) + fItem(x).

Q2:

(1) first step is to calculate the Gini index for Gender, Smokes, Chest pain, Cough, Lung Cancer:

1. calculating the Gini index for ‘Gender’:

P(Gender = Female): 2/6

P(Gender = Male): 4/6

P(Gender = Female & Lung Cancer = Yes): 1/2

P(Gender = Female & Lung Cancer = No): 1/2

Gini index for ‘Female’ of Gender = 1 - ((1/2)^2 + (1/2)^2) = 1/2

P(Gender = Male & Lung Cancer = Yes): 3/4

P(Gender = Male & Lung Cancer = No): 1/4

Gini index for ‘Male’ of Gender= 1 - ((3/4)^2 + (1/4)^2) = 5/8

Gini index for Gender = 2/6 * 1/2 + 4/6 * 5/8 = 7/12 = 0.58

2, calculating the Gine index for ‘Smokes’:

P(Smokes = Yes): 3/6

P(Smokes = No): 3/6
P(Smokes = Yes & Lung Cancer = Yes):1
P(Smokes = Yes & Lung Cancer = No):0
Gini index for 'Yes' of Smokes= 1 - ((1)^2 + (0)^2) = 0
P(Smokes = No & Lung Cancer = Yes):1/3
P(Smokes = No & Lung Cancer = No):2/3
Gini index for 'No' of Smokes = 1 - ((1/3)^2 + (2/3)^2) = 4/9
Gini index for Smokes = 3/6 * 0 + 3/6 * 4/9 = 2/9 = 0.22

3. calculating the Gini index for 'Chest pain':
P(Chest pain = Yes):4/6
P(Chest pain = No):2/6
P(Chest pain = Yes & Lung Cancer = Yes): 2/4
P(Chest pain = Yes & Lung Cancer = No): 2/4
Gini index for 'Yes' of Chest pain= 1 - ((2/4)^2 + (2/4)^2) = 1/2
P(Chest pain = No & Lung Cancer = Yes): 1
P(Chest pain = No & Lung Cancer = No): 0
Gini index for 'No' of Chest pain= 1 - ((1)^2 + (0)^2) = 0 Gini index for Chest pain = 4/6 * 1/2 + 2/6 * 0 = 1/3 = 0.33

4. calculating the Gini index for 'Cough':
P(Cough = Yes): 4/6
P(Cough = No): 2/6
P(Cough = Yes & Lung Cancer = Yes): 2/4
P(Cough = Yes & Lung Cancer = No): 2/4
Gini index for 'Yes' of Cough = 1 - ((2/4)^2 + (2/4)^2) = 1/2
P(Cough = No & Lung Cancer = Yes): 1
P(Cough = No & Lung Cancer = No): 0
Gini index for 'No' of Cough= 1 - ((1)^2 + (0)^2) = 0
Gini index for Cough = 4/6 * 1/2 + 2/6 * 0 = 0.33 We can draw the table:

Attributes	Gini index
Gender	0.58
Smokes	0.22
Chest pain	0.33
Cough	0.33

The Smokes has the lowest Gini index, so it will be the root node for decision tree.

For smokes, if smokes is yes then the lung cancer is yes, then, calculating the 'No' of smokes, so the table is:

Smokes	Gender	Chest pain	Cough	Lung Cancer
No	Male	No	No	Yes
No	Female	Yes	Yes	No

No	Male	Yes	Yes	No

1. calculating the Gini index of Gender for No Smokes:

P(Gender = Male): 2/3

P(Gender = Female): 1/3

P(Gender = Male & Lung Cancer = Yes): 1/2

P(Gender = Male & Lung Cancer = No): 1/2

Gini index for 'Male' of Gender= 1- ((1/2)^2 + (1/2)^2) = 1/2

P(Gender = Female & Lung Cancer = Yes): 0

P(Gender = Female & Lung Cancer = No): 1

Gini index for 'Female' of Gender= 1- ((0)^2 + (1)^2) = 0

Gini index for Gender = 2/3 * 1/2 + 1/3 * 0 = 1/3 = 0.33

2. calculating the Gini index of Chest pain for No Smokes:

P(Chest pain = Yes): 2/3

P(Chest pain = No): 1/3

P(Chest pain = Yes & Lung Cancer = Yes): 0

P(Chest pain = Yes & Lung Cancer = No): 1

Gini index for 'Yes' of Chest pain= 1- ((0)^2 + (1)^2) = 0

P(Chest pain = No & Lung Cancer = Yes): 1

P(Chest pain = No & Lung Cancer = No): 0

Gini index for 'No' of Chest pain= 1 - ((0)^2 + (1)^2) = 0

Gini index for Chest pain = 2/3 * 0 + 1/3 * 0 = 0

3. calculating the Gini index of Cough for No Smokes:

P(Cough = Yes): 2/3

P(Cough = No): 1/3

P(Cough = Yes & Lung Cancer = Yes): 0

P(Cough = Yes & Lung Cancer = No): 1

Gini index for 'Yes' of Cough= 1- ((0)^2 + (1)^2) = 0

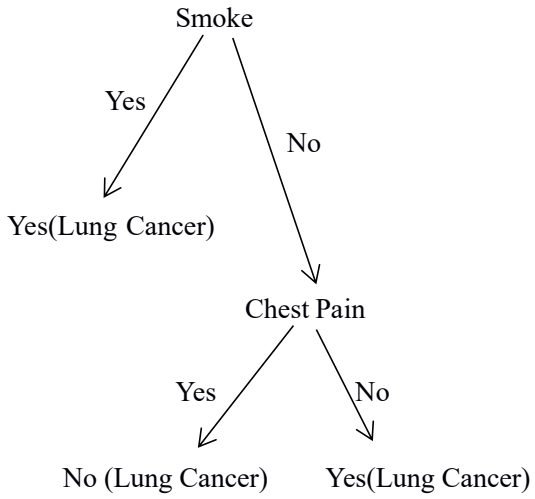
P(Cough = No & Lung Cancer = Yes): 1

P(Cough = No & Lung Cancer = No): 0

Gini index for 'No' of Cough= 1 - ((0)^2 + (1)^2) = 0 Gini index for Cough = 2/3 * 0 + 1/3 * 0 = 0

From the above, we can see the Gini index for Chest pain and Cough is the lowest and they are the same which is 0. So, we can infer that the next node is Chest pain or Cough, and the Gender is no relate element and it will not affect the the result of Lung Cancer.

So, the decision tree has two possible situations and I choose one situation which is Chest pain and I draw it below:



(2) Translate the decision tree to decision rules:

From question(1), we can see the two possible decision trees,then translating them to decision rules, the rules are blew:

IF Smokes = “Yes” THEN Lung Cancer = “Yes”

IF Smokes = “No” AND Chest pain = “Yes” THEN Lung Cancer = “No”

IF Smokes = “No” AND Chest pain = “No” THEN Lung Cancer = “Yes”

Q3:

(1) From the topic, we can know that class attribute y only takes two values: 1 or 0, also,there is a additional assumptions that x is a binary vector (only takes 0 or 1).

The Bayesian theorem: $$P(h|X) = \frac{P(X|h)P(h)}{P(X)}$$

And it can be written as posterior = likelihood * prior / evidence

So, if $\frac{P(y=1|x)}{P(y=0|x)} > 1$, then the y is 1,else the y is 0

Then the log-likelihood can be $log(\frac{P(y=1|x)}{P(y=0|x)}) > 0$,then y is 1,else y is 0

From the Bayesian theorem formula, the above formula can be change into:

$$log(\frac{P(x|y=1) * P(y=1)}{P(x|y=0) * P(y=0)}) > 0$$

Then the likelihood of the whole training dataset is:

$$log(\frac{P(y=1)}{P(y=0)}) + log(\frac{P(x_1|y=1)*P(x_2|y=1)*P(x_3|y=1)*\cdots*P(x_d|y=1)}{P(x_1|y=0)*P(x_2|y=0)*P(x_3|y=0)*\cdots*P(x_d|y=0)}) > 0$$

=
$$log(\frac{P(y=1)}{P(y=0)}) + log\prod_{i=0}^{d}(\frac{P(x_i|y=1)}{P(x_i|y=0)}) > 0$$

Because of
$$log\prod_{i} x_i = \sum_{i} logx_i$$

So, the formula can change to:

$$log(\frac{P(y=1)}{P(y=0)}) + \sum_{i=0}^{d} log(\frac{P(x_i|y=1)}{P(x_i|y=0)}) > 0$$

For this part $log(\frac{P(x_i|y=1)}{P(x_i|y=0)})$, assuming it as $t(i, x_i)$ from the 1 to d dimensions

Then, the later part formula can be changed into:

$$\sum_{i=1}^{d} t_i = \sum_{i=1}^{d}(t(i,1)*x_i + t(i,0)(1-x_i))$$

$$= \sum_{i=1}^{d}(t(i,1)*x_i - x_i*t(i,0) + t(i,0))$$

$$= \sum_{i=1}^{d}((t(i,1)-t(i,0))*x_i + t(i,0))$$

$$= \sum_{i=1}^{d} t(i,0) + \sum_{i=1}^{d}(t(i,1)-t(i,0))*x_i$$

Then the whole formula can be:

$$log(\frac{P(y-1)}{P(y-0)}) + \sum_{i=1}^{d} t(i,0) + \sum_{i=1}^{d}(t(i,1)-t(i,0))*x_i$$

$$log(\frac{P(y-1)}{P(y-0)}) + \sum_{i=1}^{d} t(i,0)$$

This part $t(i,1)-t(i,0)$ as w_i Assuming this part as a

Then the formula can be:

$$a + \sum_{i=1}^{d} w_i * x_i$$

for the i which is from i to d, unfolding the w and x:

X is 1,2,3,...,d

$$w^T = [a, w_1, w_2, w_3, ..., w_d]$$

For the a, we can think it is w_{d+1}

So, for d+1-dimension space, the naive bayes classifier is a linear classifier

(2) From the above first question process, we can know that learning the w of naive bayes, we just need to compute the $log(\frac{P(x_i|y=1)}{P(x_i|y=0)})$ from i to d once.

But for logistic regression, firstly, we have to use sigmoid function, then for the maximize the log-likelihood and it is a concave, it will also takes a partial derivatives. There are more steps of LR than NB.

From the above procedures, we can know that the learning W of naive bayes is much easier than learning W of logistic regression.