COMP9318 - Assignment 1

Yao Yuan z5092195

Q1. (1)

	Location	Time	Item	SUM(Quantity)
0	ALL	2005	ALL	3100
1	ALL	2005	PS2	1400
2	ALL	2005	XBox 360	1700
3	ALL	2006	ALL	2000
4	ALL	2006	PS2	1500
5	ALL	2006	Wii	500
6	ALL	ALL	ALL	5100
7	ALL	ALL	PS2	2900
8	ALL	ALL	Wii	500
9	ALL	ALL	XBox 360	1700
10	Melbourne	2005	ALL	1700
11	Melbourne	2005	XBox 360	1700
12	Melbourne	ALL	ALL	1700
13	Melbourne	ALL	XBox 360	1700
14	Sydney	2005	ALL	1400
15	Sydney	2005	PS2	1400
16	Sydney	2006	ALL	2000
17	Sydney	2006	PS2	1500
18	Sydney	2006	Wii	500
19	Sydney	ALL	ALL	3400
20	Sydney	ALL	PS2	2900

	Location	Time	Item	SUM(Quantity)
21	Sydney	ALL	Wii	500

Q1. (2)

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

Q1. (3)

	Location	Time	Item	SUM(Quantity)
0	ALL	2005	ALL	3100
1	ALL	2006	ALL	2000
2	ALL	ALL	PS2	2900
3	Sydney	2006	ALL	2000
4	Sydney	ALL	ALL	3400
5	Sydney	ALL	PS2	2900
6	ALL	ALL	ALL	5100

Q1. (4)

$$f(x) = 16 * f_{Location}(x) + 4 * f_{Time}(x) + f_{Item}(x)$$

This function is feasible.
Reason:
For MOLAP, we have to make sure two different combination does not end into the same result.
However, for the first question, an easy counter example is that:
The result of function for combination of “ALL-2006-Wii” = 9
The result of function for combination of “Sydney-ALL-ALL”= 9
These two have the same function result but they should be different.

MOLAP cube:

	Offset	SUM(Quantity)
0	4	3100
1	5	1400
2	6	1700
3	8	2000
4	9	1500
5	11	500

	Offset	SUM(Quantity)
6	0	5100
7	1	2900
8	3	500
9	2	1700
10	36	1700
11	38	1700
12	32	1700
13	34	1700
14	20	1400
15	21	1400
16	24	2000
17	25	1500
18	27	500
19	16	3400
20	17	2900
21	19	500

Q2. (1)

From the lecture we know that

$$\text{gini(T)} = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j^2 is the relative frequency of class j in T.

if we split the dataset into two subset, we can calculate the gini index using the following formula:

$$gini_{split}(\text{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

In our cases, we have two classes having cancer or not having cancer.

1. If we do not split any attribute:

Total instances = 6
Total number of patients having Cancer = 4
Total number of patients not having Cancer = 2
Therefore, after calculation

$$\text{gini(T)} = \frac{4}{9}$$

2. If we split upon Gender:

Gender	Yes, having cancer	No, not having cancer
Male	3	1
Female	1	1

$$gini_{split}(\text{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\text{T}) = \frac{4}{6} gini(T_1) + \frac{2}{6} gini(T_2)$$

$$gini_{split}(\text{T}) = \frac{4}{6} * \left(1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right)\right) + \frac{2}{6} * \left(1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right)\right)$$

$$gini_{split}(\text{T}) = \frac{5}{12} = 0.4167$$

3. If we split upon Smokes:

Smokes	Yes, having cancer	No, not having cancer
Yes	3	0
No	1	2

$$gini_{split}(\mathrm{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{3}{6} gini(T_1) + \frac{3}{6} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{3}{6} * \left(1 - \left(\left(\frac{3}{3}\right)^2 + (0)^2\right)\right) + \frac{3}{6} * \left(1 - \left(\left(\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2\right)\right)$$

$$gini_{split}(\mathrm{T}) = \frac{2}{9} = 0.2222$$

4. If we split upon Chest pain:

Chest pain	Yes, having cancer	No, not having cancer
Yes	2	2
No	2	0

$$gini_{split}(\mathrm{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{4}{6} gini(T_1) + \frac{2}{6} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{4}{6} * \left(1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right)\right) + \frac{2}{6} * \left(1 - \left(\left(\frac{2}{2}\right)^2 + (0)^2\right)\right)$$

$$gini_{split}(\mathrm{T}) = \frac{1}{3} = 0.3333$$

5. If we split upon Cough:

Cough	Yes, having cancer	No, not having cancer
Yes	2	2
No	2	0

$$gini_{split}(\mathrm{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{4}{6} gini(T_1) + \frac{2}{6} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{4}{6} * \left(1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right)\right) + \frac{2}{6} * \left(1 - \left(\left(\frac{2}{2}\right)^2 + (0)^2\right)\right)$$

$$gini_{split}(\mathrm{T}) = \frac{1}{3} = 0.3333$$

Therefore, we should firstly split on Smokes because it has the smallest GINI index.
After splitting, we can get two datasets:

For dataset, when 'Smokes' = 'Yes'

Patient ID	Gender	Chest pain	Cough	Lung Cancer
1	Female	Yes	Yes	Yes
2	Male	No	Yes	Yes
5	Male	Yes	No	Yes

If we do not split any attribute:
Total instances = 3
Total number of patients having Cancer = 3
Total number of patients not having Cancer = 0
Therefore, after calculation

$$\text{gini(T)} = 0$$

since gini index is always non-negative, we conclude that in this dataset, we do not need to split any attributes.

For dataset, when 'Smokes' = 'No'

Patient ID	Gender	Chest pain	Cough	Lung Cancer
3	Male	No	No	Yes
4	Female	Yes	Yes	No
6	Male	Yes	Yes	No

1. If we do not split any attribute:

Total instances = 3
Total number of patients having Cancer = 1
Total number of patients not having Cancer = 2
Therefore, after calculation

$$\text{gini(T)} = \frac{4}{9}$$

2. If we split upon Gender:

Gender	Yes, having cancer	No, not having cancer
Male	1	1
Female	0	1

$$gini_{split}(\text{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\text{T}) = \frac{2}{3} gini(T_1) + \frac{1}{3} gini(T_2)$$

$$gini_{split}(\text{T}) = \frac{2}{3} * \left(1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right)\right) + \frac{1}{3} * \left(1 - \left(\left(\frac{1}{1}\right)^2 + (0)^2\right)\right)$$

$$gini_{split}(\mathrm{T}) = \frac{1}{3}$$

3. If we split upon Chest pain:

Chest pain	Yes, having cancer	No, not having cancer
Yes	0	2
No	1	0

$$gini_{split}(\mathrm{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{2}{3} gini(T_1) + \frac{1}{3} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{2}{3} * \left(1 - \left((0)^2 + (\frac{2}{2})^2\right)\right) + \frac{1}{3} * (1 - \left((\frac{1}{1})^2 + (0)^2\right))$$

$$gini_{split}(\mathrm{T}) = 0$$

4. If we split upon Cough:

Cough	Yes, having cancer	No, not having cancer
Yes	0	2
No	1	0

$$gini_{split}(\mathrm{T}) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{2}{3} gini(T_1) + \frac{1}{3} gini(T_2)$$

$$gini_{split}(\mathrm{T}) = \frac{2}{3} * \left(1 - \left((0)^2 + (\frac{2}{2})^2\right)\right) + \frac{1}{3} * (1 - \left((\frac{1}{1})^2 + (0)^2\right))$$

$$gini_{split}(\mathrm{T}) = 0$$

Therefore, we can see that both 'Cough' and 'Chest pain' have gini index equals to 0.
We can randomly pick one.
At here, we pick Chest pain.

After we split again on 'Chest pain',
We have two split datasets:

When Chest pain is No:

Patient ID	Gender	Cough	Lung Cancer
3	Male	No	Yes

Total instances = 1
Total number of patients having Cancer = 1
Total number of patients not having Cancer = 0
Therefore, after calculation

$$\text{gini}(\text{T}) = 0$$

When Chest pain is Yes:

Patient ID	Gender	Cough	Lung Cancer
4	Female	Yes	No
6	Male	Yes	No

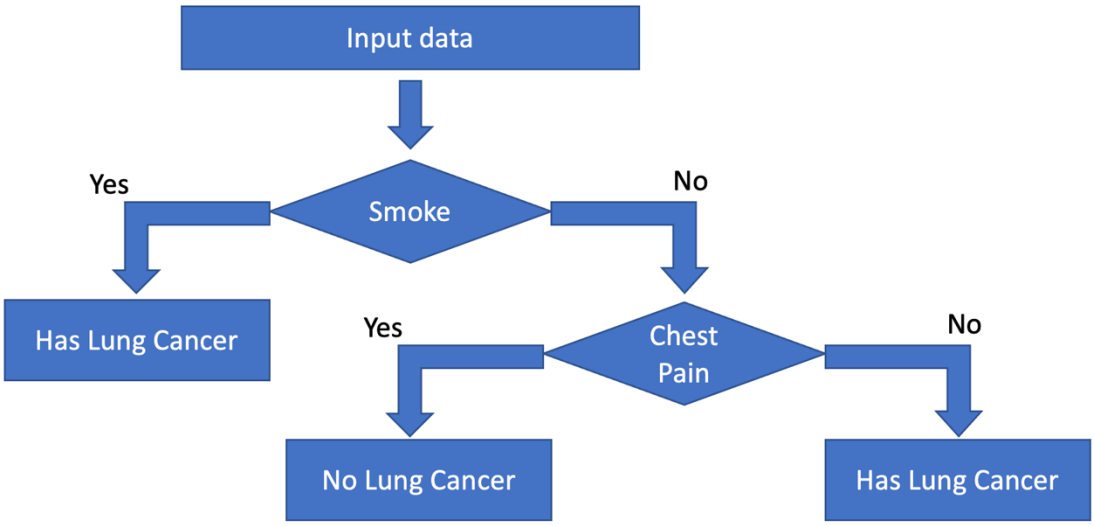
Total instances = 2
Total number of patients having Cancer = 0
Total number of patients not having Cancer = 2
Therefore, after calculation

$$\text{gini}(\text{T}) = 0$$

Since both gini index equal to 0, there is no need to split.

Q2. (2)

```
If Input_data["Smoke"] == 'Yes':
        Return "Has Lung Cancer"
Else if Input_data["Smoke"] == 'No':
        If Input_data["Chest Pain"] == 'Yes':
                Return "No Lung Cancer"
        Else if Input_data["Chest Pain"] == 'No':
                Return "Has Lung Cancer"
```



Q3. (1)

For Naïve Bayes classifier, we know that, given a test instance $\vec{x}$ with d-dimensions, it belongs to class C_1 if $P_r(C_1|\vec{x}) > P_r(C_0|\vec{x})$, class C_2 if $P_r(C_1|\vec{x}) < P_r(C_0|\vec{x})$ and either class if $P_r(C_1|\vec{x}) = P_r(C_0|\vec{x})$.

Since we have add-one smooth, we won't have $P_r(C_1|\vec{x}) = 0 \; or \; P_r(C_0|\vec{x}) = 0$.
Let's assume a log function $g(\vec{x})$, such that

$$g(\vec{x}) = log\frac{P_r(C_1|\vec{x})}{P_r(C_0|\vec{x})}$$

And class of $\vec{x}$:

$$C(\vec{x}) = \begin{cases} C_1, & if\ g(\vec{x}) > 0 \\ C_0, & if\ g(\vec{x}) < 0 \end{cases}$$

From the Bayesian Theorem, we can transform $g(\vec{x})$ as:

$$g(\vec{x}) = log\frac{P_r(\vec{x}|C_1) * P_r(C_1)}{P_r(\vec{x}|C_0) * P_r(C_0)}$$

$$g(\vec{x}) = \log\left(\frac{P_r(\vec{x}|C_1)}{P_r(\vec{x}|C_0)} * \frac{P_r(C_1)}{P_r(C_0)}\right)$$

$$g(\vec{x}) = log\frac{P_r(\vec{x}|C_1)}{P_r(\vec{x}|C_0)} + log\frac{P_r(C_1)}{P_r(C_0)}$$

For Naïve Bayes, we have conditionally independent assumption on attributes, so we can write $g(\vec{x})$ as:

$$g(\vec{x}) = log\frac{\prod_{i=1}^{d} P_r(x_i|C_1)}{\prod_{i=1}^{d} P_r(x_i|C_0)} + log\frac{P_r(C_1)}{P_r(C_0)}$$

$$g(\vec{x}) = \sum_{i=1}^{d} log\frac{P_r(x_i|C_1)}{P_r(x_i|C_0)} + log\frac{P_r(C_1)}{P_r(C_0)}$$

From the question, we know that x can either be 0 or 1:

$$g(\vec{x}) = \sum_{i=1}^{d} log\frac{P_r(x_i = 1|C_1) * x_i + P_r(x_i = 0|C_1) * (1 - x_i)}{P_r(x_i = 1|C_0) * x_i + P_r(x_i = 0|C_0) * (1 - x_i)} + log\frac{P_r(C_1)}{P_r(C_0)}$$

From this equation, we can see that $P_r(x_i = 1|C_1)$ is the $P_r(1|C_1)$ at attribute i.

Therefore, by looking at the pattern at the above formula, we can assume

$$f(i, x_i) = log\frac{P_r(x_i = 1|C_1) * x_i + P_r(x_i = 0|C_1) * (1 - x_i)}{P_r(x_i = 1|C_0) * x_i + P_r(x_i = 0|C_0) * (1 - x_i)}$$

And lets also assume a constant τ, such that

$$\tau = log\frac{P_r(C_1)}{P_r(C_0)}$$

Therefore, we can rewrite $g(\vec{x})$ as:

$$g(\vec{x}) = \sum_{i=1}^{d} f(i, x_i) + \tau$$

$$g(\vec{x}) = \sum_{i=1}^{d} (f(i,1) * x_i + f(i,0) * (1 - x_i)) + \tau$$

$$g(\vec{x}) = \sum_{i=1}^{d} f(i,0) + \sum_{i=1}^{d} (f(i,1) - f(i,0)) * x_i + \tau$$

Therefore, we can let

$$w_i = f(i,1) - f(i,0)$$

$$\alpha = \sum_{i=1}^{d} f(i,0) + \tau$$

$$g(\vec{x}) = \sum_{i=1}^{d} w_i * x_i + \alpha$$

Where $f(i,0)$ is corresponding to each attribute.

Therefore, we can show that Naïve Bayes is a linear classifier, and it has a dummy attribute in its feature space which forms a d+1-dimension feature space.

And $\vec{w} = [\alpha, w_1, w_2, \dots, w_d]$.

Q3. (2)

From Q3.(1), we know that to calculate the parameter $\vec{w}$, we just need to calculate the count of attribute value 1 and 0 in each class for each attribute, which can be easily computed. However, for Logistic regression, there is no closed-form solution to maximize the likelihood. For example, we can use Gradient Ascent algorithm to find parameter $\vec{w}$ that maximize the likelihood which requires larger computation force. Therefore, learning w_{NB} is much easier than learning w_{LR}.