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 |
| 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)

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 |
| 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

where 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:

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

1. If we split upon Gender:

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

1. If we split upon Smokes:

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

1. If we split upon Chest pain:

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

1. If we split upon Cough:

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

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

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

1. If we split upon Gender:

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

1. If we split upon Chest pain:

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

1. If we split upon Cough:

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

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

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”

Diagram

Description automatically generated

Q3. (1)

For Naïve Bayes classifier, we know that, given a test instance with d-dimensions, it belongs to class if , class if and either class if .

Since we have add-one smooth, we won’t have .

Let’s assume a log function , such that

And class of :

From the Bayesian Theorem, we can transform as:

For Naïve Bayes, we have conditionally independent assumption on attributes, so we can write as:

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

From this equation, we can see that is the at attribute i.

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

And lets also assume a constant τ, such that

Therefore, we can rewrite as:

Therefore, we can let

Where 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 .

Q3. (2)

From Q3.(1), we know that to calculate the parameter , 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 that maximize the likelihood which requires larger computation force. Therefore, learning

is much easier than learning .