**Data Mining Assignment 4**

[**Classification Basics**](https://learn.rochester.edu/webapps/assignment/uploadAssignment?content_id=_5784093_1&course_id=_71221_1&group_id=&mode=view)

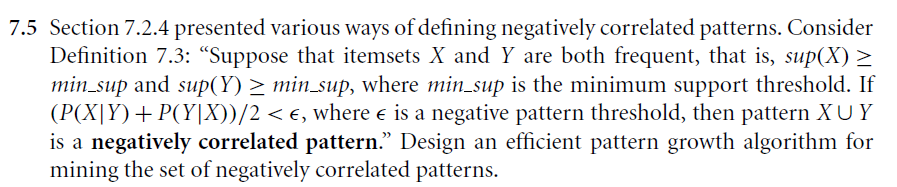
-  7.5\*, 7.9, 7.10\*

-  8.3, 8.5, 8.7, 8.12, 8.14

-  Due 10/13

\* changed from previous years

**Aradhya Mathur**



**Answer)**

Using FP Growth, we generate all frequent itemsets from the database. Both frequent and infrequent patterns have to be stored.

We have to find P(X|Y) and P(Y|X)

We know P(A|B) = P(A) / P(B)

Similarly, P(X|Y) = P( X Y) / P(Y) and P(Y|X) = P(X Y) / P(X)

Now we calculate P(X|Y) and P(Y|X)

P(X Y) = P(Z)

So, we write sup(X), sup(Y) and sup(Z) for P(X), P(Y) and P(Z) respectively.

P(X/Y) = P( X Y) / P(Y) = sup(Z) / sup(Y)

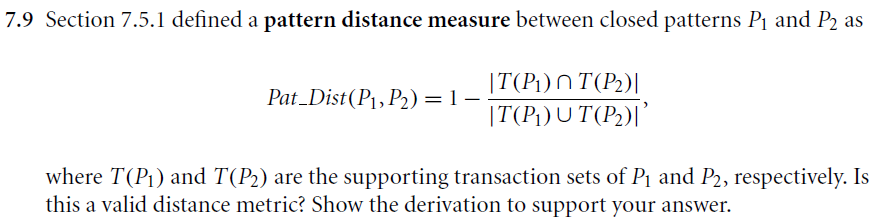
Similarly, P(Y/X) = P( X Y) / P(X) = sup(Z) / sup(X)

Now ( P(X|Y )+P(Y|X) ) / 2 can be written as

= ( sup(Z)/sup(Y) + sup(Z)/sup(X) )/2

If ( sup(Z)/sup(Y) + sup(Z)/sup(X) )/2 < , where is negative pattern threshold.

Using this we can calculate if X Y are negatively correlated pattern.



**Answer)**

P1 and P2 are two closed patterns

T(P1) and T(P2) are supporting transaction sets.

Pat\_Dist (P1,P2) = 1-

There are four properties to be satisfied:

1. Dist(P1, P2) > 0 for all P1 ≠ P2

We know, >

< 1

> 0, Hence satisfied

1. Dist(P1, P2) = 0 for all P1 = P2

If P1 = P2 then are same and hence numerator = denominator in

This leads to Dist(P1,P2) = 1-1 = 0, Hence satisfied

1. Dist(P1, P2) = Dist(P2, P1)’

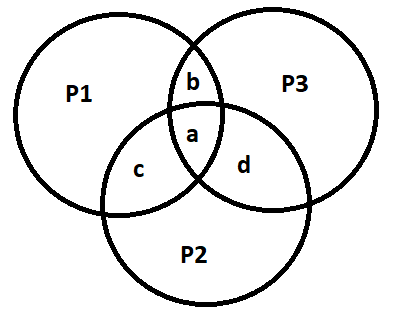
As = and =

= , Hence satisfied

1. Dist(P1, P2) + Dist(P2, P3) ≥ Dist(P1, P3) for all P1, P2, P3

Basic triangle law which states that sum of any two sides should be greater than the 3rd side.

Using venn diagram here,



|T(P1) ∩ T(P2)| + |T(P1) ∩ T(P3)| −|T(P1) ∩ T(P2) ∩ T(P3)| ≤ |T(P1)|

c + b – a ≤ T(P1) this can be easily seen and interpreted through venn diagram. As c+b-a will be part of whole P1, at most it can be equal if there are no distinct elements. This holds for P2 and P3 as well.

We can show this in the form,

1 - + 1 - ≥ 1 -

= (Basic Set Property)

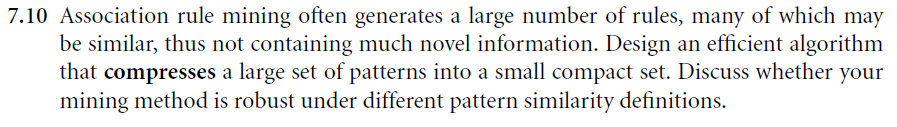
1 - + 1 -

2 - - ≥ Dist(P1, P3)

This proves that Dist(P1, P2) + Dist(P2, P3) ≥ Dist(P1, P3) , Hence satisfied

All four properties are satisfied

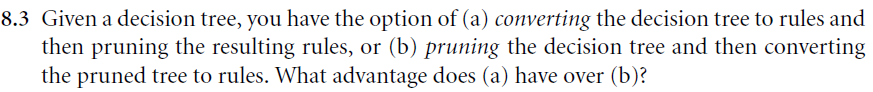
Therefore, Pat\_Dist (P1,P2) – 1 - is valid distance metric.



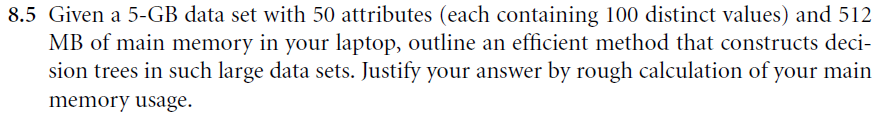
**Answer)** Pattern clustering can be used. When we mine association rules from non-transactional data, we might obtain large number of rules. Some of them might not be required or some might be redundant. Using clustering we can efficiently generate association mining rules. We can form general rules for clustering. For example, all the association rules (work\_hours = 21) and (full\_time = yes) till (work\_hours = 39) and (full\_time = yes) can be written as (20 < work\_hours < 40) =) and (full\_time = yes) as per new clustered rules. Clustered rules are useful in reducing huge number of association rules. These rules are easy to understand.

Also, we can use top-k most frequent patterns but it is better to use k most interesting patterns. In k most interesting, patterns are mutually independent, have less redundancy. There are patterns which have high significance along redundancy are called redundancy-aware top-k patterns but there is a trade-off between redundancy and significance.

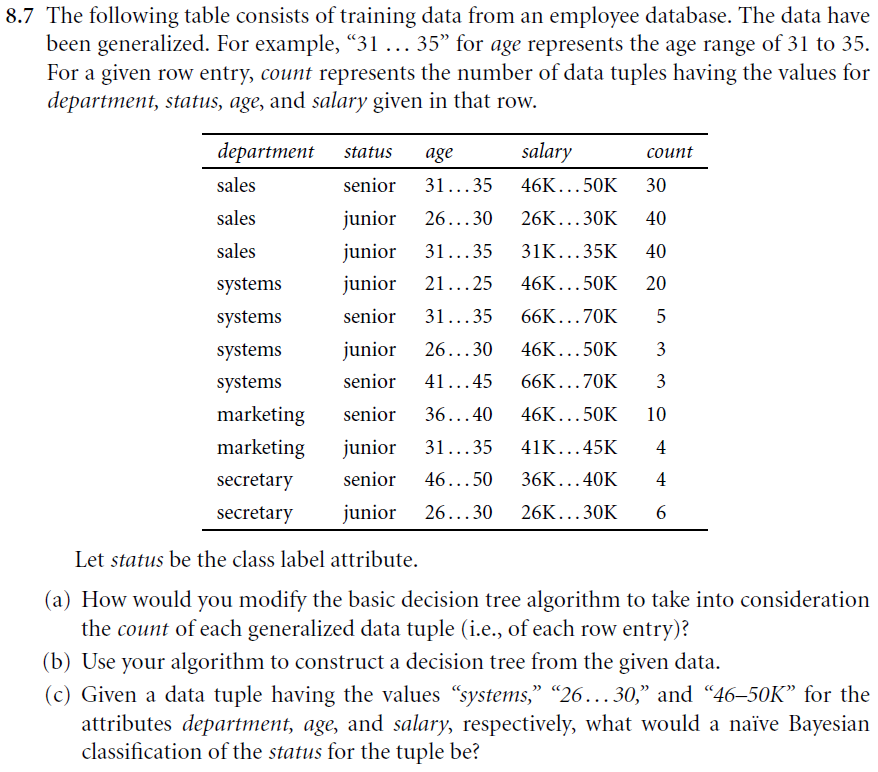
None of these methods are robust in nature.



**Answer)** Issue of overfitting is resolved by pruning. Instead of pruning a complete decision node, we can prune a single path as each path will have a different rule. In case root nodes are removed, pruning can occur without considering rebuilding of the tree. Also, rules are simpler to understand. Therefore (a) is better than (b)



**Answer)** This can be solved using RainForest. We will be creating AVC lists. AVC means Attribute-Value, Classlabel. There are total of 50 attributes and for each attribute there will be a AVC list of size 100xC as 100 distinct values are given. 50x100xC will be the total size as there are 50 attributes in total. 512 MB is more than enough to accommodate AVC sets for some value of C. In this we have to take care of units for example, 1 MB = 106 Bytes and 1 Byte = 8 bits. Therefore 512 MB is equal to 4096 x 106 bits. So, it is very huge for accommodating 50x100xC for any value of C. C will be in bits. And according to our memory maximum value of C can be 819,200 bits which is more than enough.



**Answer)**

1. While calculating information gain, count of each tuple has to be taken into account. Count is also to find most common class.
2. **Method 1) Remove count column**

|  |  |  |  |
| --- | --- | --- | --- |
| **Department** | **Age** | **Salary** | **Status** |
| Sales | 31..35 | 46-50 | Senior |
| Sales | 26..30 | 26-30 | Junior |
| Sales | 31..35 | 31-35 | Junior |
| Systems | 21..25 | 46-50 | Junior |
| Systems | 31..35 | 66-70 | Senior |
| Systems | 26..30 | 46-50 | Junior |
| Systems | 41..45 | 66-70 | Senior |
| Marketing | 36..40 | 46-50 | Senior |
| Marketing | 31..35 | 41-45 | Junior |
| Secretary | 46..50 | 36-40 | Senior |
| Secretary | 26..30 | 26-30 | Junior |

Senior = P = 5

Junior = N = 6

Entropy = - log () - log ()

Using base 2 for log function

Entropy = - log () - log ()

Entropy = 0.994

**For Department**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| Sales | 1 | 2 | 0.91833 |
| Systems | 2 | 2 | 1 |
| Marketing | 1 | 1 | 1 |
| Secretary | 1 | 1 | 1 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Department =

= (3\*0.91833 + 4\*1 + 2\*1 +2\*1) / 11 = 0.9777

Gain = Entropy – Entropy of Department = 0.994-0.9777 = 0.0163

**For Age**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| 21..25 | 0 | 1 | 0 |
| 26..30 | 0 | 3 | 0 |
| 31..35 | 2 | 2 | 1 |
| 36..40 | 1 | 0 | 0 |
| 41..45 | 1 | 0 | 0 |
| 46..50 | 1 | 0 | 0 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Age =

= 4\*1 / 11 = 0.36363

Gain = Entropy – Entropy of Age = 0.994-0.3636 = 0.6304

**For Salary**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| 26..30 | 0 | 2 | 0 |
| 31..35 | 0 | 1 | 0 |
| 36..40 | 1 | 0 | 0 |
| 41..45 | 0 | 1 | 0 |
| 46..50 | 2 | 2 | 1 |
| 66-70 | 2 | 0 | 0 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Salary =

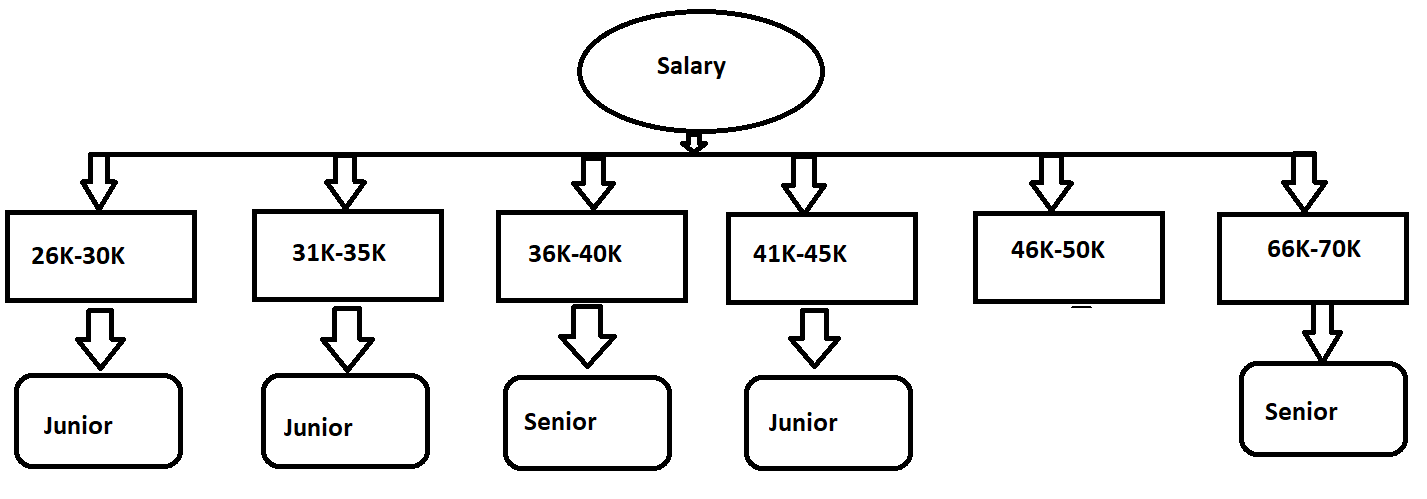
= 4\*1 / 11 = 0.36363

Gain = Entropy – Entropy of Salary = 0.994-0.3636 = 0.6304

Gain of both Salary and Age are same

Let’s take Salary as root note

**Step** **1)**



**For Salary 46K-50K**

Table looks like

|  |  |  |
| --- | --- | --- |
| Department | Age | Status |
| Sales | 31..35 | S |
| Systems | 21..25 | J |
| Systems | 26..30 | J |
| Marketing | 36..40 | S |

Similarly repeating the steps

Senior = P = 2

Junior = N = 2

Entropy = - log () - log ()

Using base 2 for log function

Entropy = - log () - log ()

Entropy = 1

**For Department**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| Sales | 1 | 0 | 0 |
| Systems | 0 | 2 | 0 |
| Marketing | 1 | 0 | 0 |
| Secretary | 0 | 0 | 0 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Department =

= 0

Gain = Entropy – Entropy of Department = 1-0 = 1

**For Age**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| 21..25 | 0 | 1 | 0 |
| 26..30 | 0 | 1 | 0 |
| 31..35 | 1 | 0 | 0 |
| 36..40 | 1 | 0 | 0 |

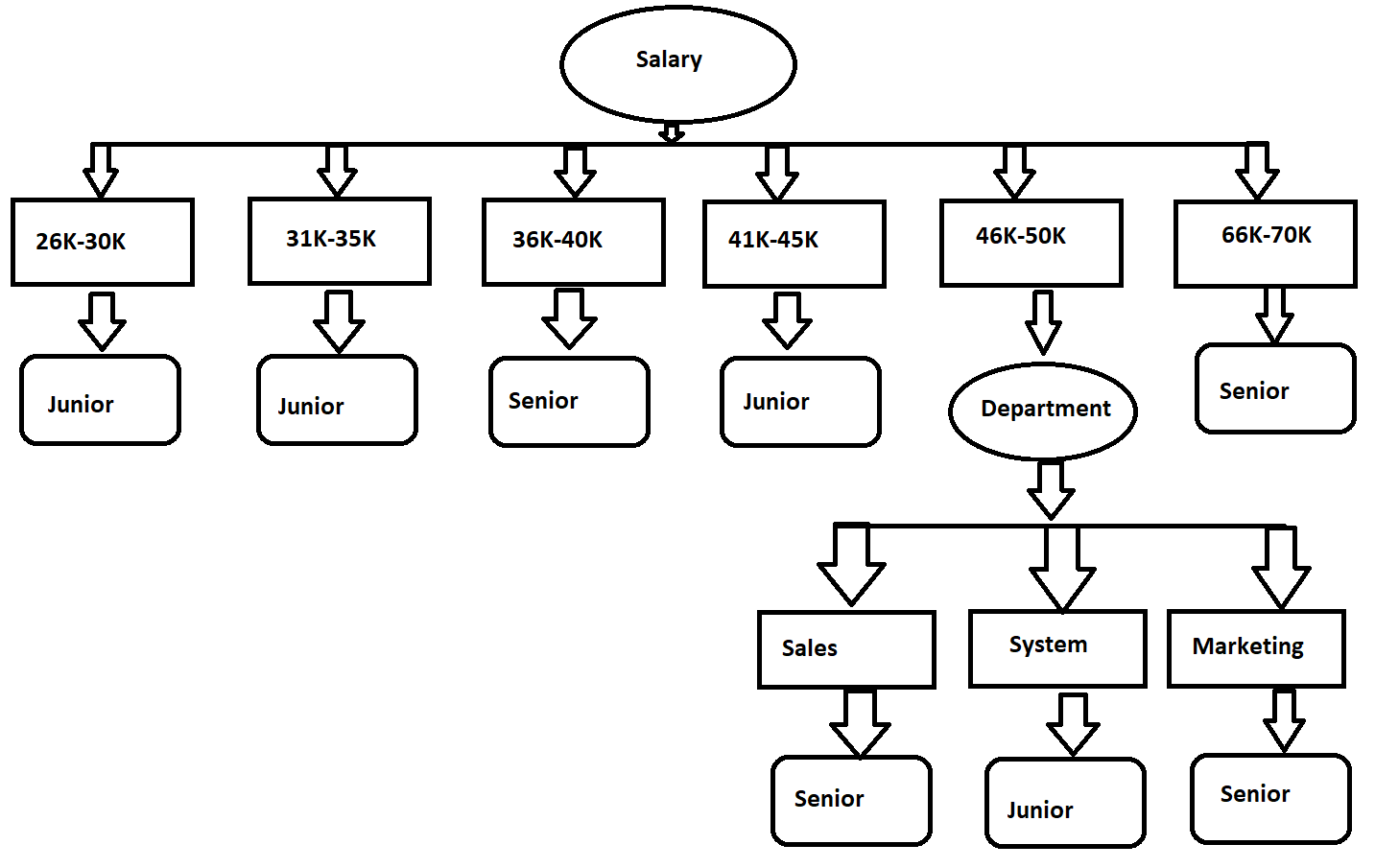
I(Pi, Ni) is calculated using = - log () - log ()

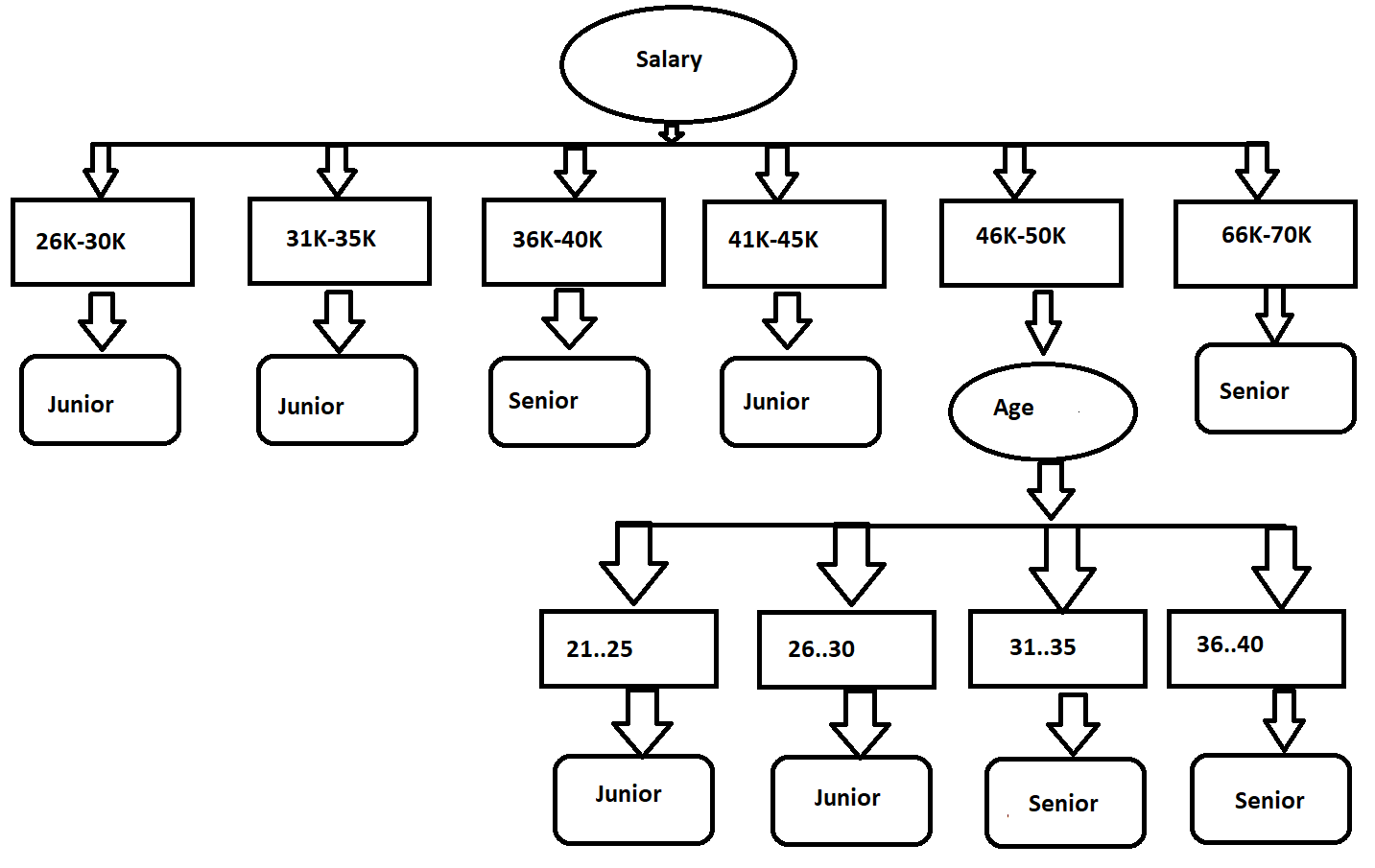
Entropy of Department =

= 0

Gain = Entropy – Entropy of Age = 1-0 = 1

**Gain is same for both**





Both decision tree will work efficiently

**Method 2) Considering count column:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department** | **Age** | **Salary** | **Count** | **Status** |
| Sales | 31..35 | 46-50 | 30 | Senior |
| Sales | 26..30 | 26-30 | 40 | Junior |
| Sales | 31..35 | 31-35 | 40 | Junior |
| Systems | 21..25 | 46-50 | 20 | Junior |
| Systems | 31..35 | 66-70 | 5 | Senior |
| Systems | 26..30 | 46-50 | 3 | Junior |
| Systems | 41..45 | 66-70 | 3 | Senior |
| Marketing | 36..40 | 46-50 | 10 | Senior |
| Marketing | 31..35 | 41-45 | 4 | Junior |
| Secretary | 46..50 | 36-40 | 4 | Senior |
| Secretary | 26..30 | 26-30 | 6 | Junior |

Senior = P = 52

Junior = N = 113

Entropy = - log () - log ()

Using base 2 for log function

Entropy = - log () - log ()

Entropy = 0.8990

**For Department**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| Sales | 30 | 80 | 0.8453 |
| Systems | 8 | 23 | 0.8237 |
| Marketing | 10 | 4 | 0.8439 |
| Secretary | 4 | 6 | 0.971 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Department =

= (110\*0.8453 + 31\*0.8237 + 14\*0.8439 +10\*0.971) / 165 = 0.8487

Gain = Entropy – Entropy of Department = 0.8990-0.8487 = 0.0503

**For Age**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| 21..25 | 0 | 20 | 0 |
| 26..30 | 0 | 49 | 0 |
| 31..35 | 35 | 44 | 0.9905 |
| 36..40 | 10 | 0 | 0 |
| 41..45 | 3 | 0 | 0 |
| 46..50 | 4 | 0 | 0 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Age =

= 79\*0.9905 / 165 = 0.4742

Gain = Entropy – Entropy of Age = 0.8990-0.4742 = 0.4248

**For Salary**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| 26..30 | 0 | 46 | 0 |
| 31..35 | 0 | 40 | 0 |
| 36..40 | 4 | 0 | 0 |
| 41..45 | 0 | 4 | 0 |
| 46..50 | 40 | 23 | 0.9468 |
| 66-70 | 8 | 0 | 0 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Salary =

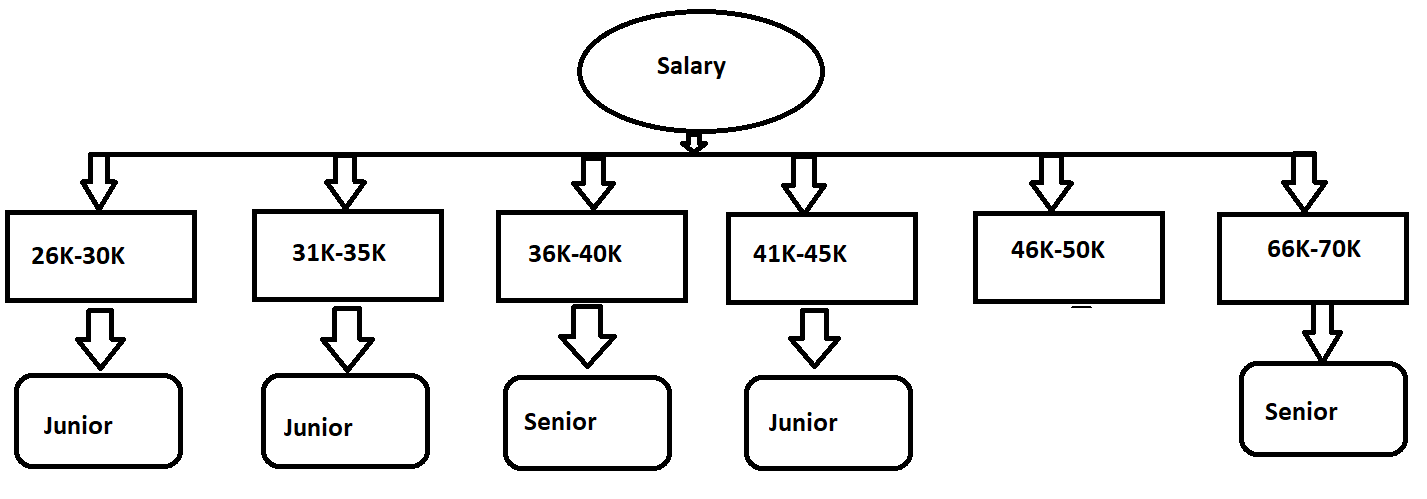
= 63\*0.9468 / 165 = 0.3615

Gain = Entropy – Entropy of Salary = 0.8990-0.3615 = 0.5375

Gain of both Salary is highest

We take Salary as root note

**Step** **1)**



**For Salary 46K-50K**

Table looks like

|  |  |  |  |
| --- | --- | --- | --- |
| Department | Age | Count | Status |
| Sales | 31..35 | 30 | S |
| Systems | 21..25 | 20 | J |
| Systems | 26..30 | 3 | J |
| Marketing | 36..40 | 10 | S |

Similarly repeating the steps

Senior = P = 40

Junior = N = 23

Entropy = - log () - log ()

Using base 2 for log function

Entropy = 0.9468

**For Department**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| Sales | 30 | 0 | 0 |
| Systems | 0 | 23 | 0 |
| Marketing | 10 | 0 | 0 |
| Secretary | 0 | 0 | 0 |

I(Pi, Ni) is calculated using = - log () - log ()

Entropy of Department =

= 0

Gain = Entropy – Entropy of Department = 0.9468-0 = 0.9468

**For Age**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Senior | Junior | I(Pi, Ni) |
| 21..25 | 0 | 20 | 0 |
| 26..30 | 0 | 3 | 0 |
| 31..35 | 30 | 0 | 0 |
| 36..40 | 10 | 0 | 0 |

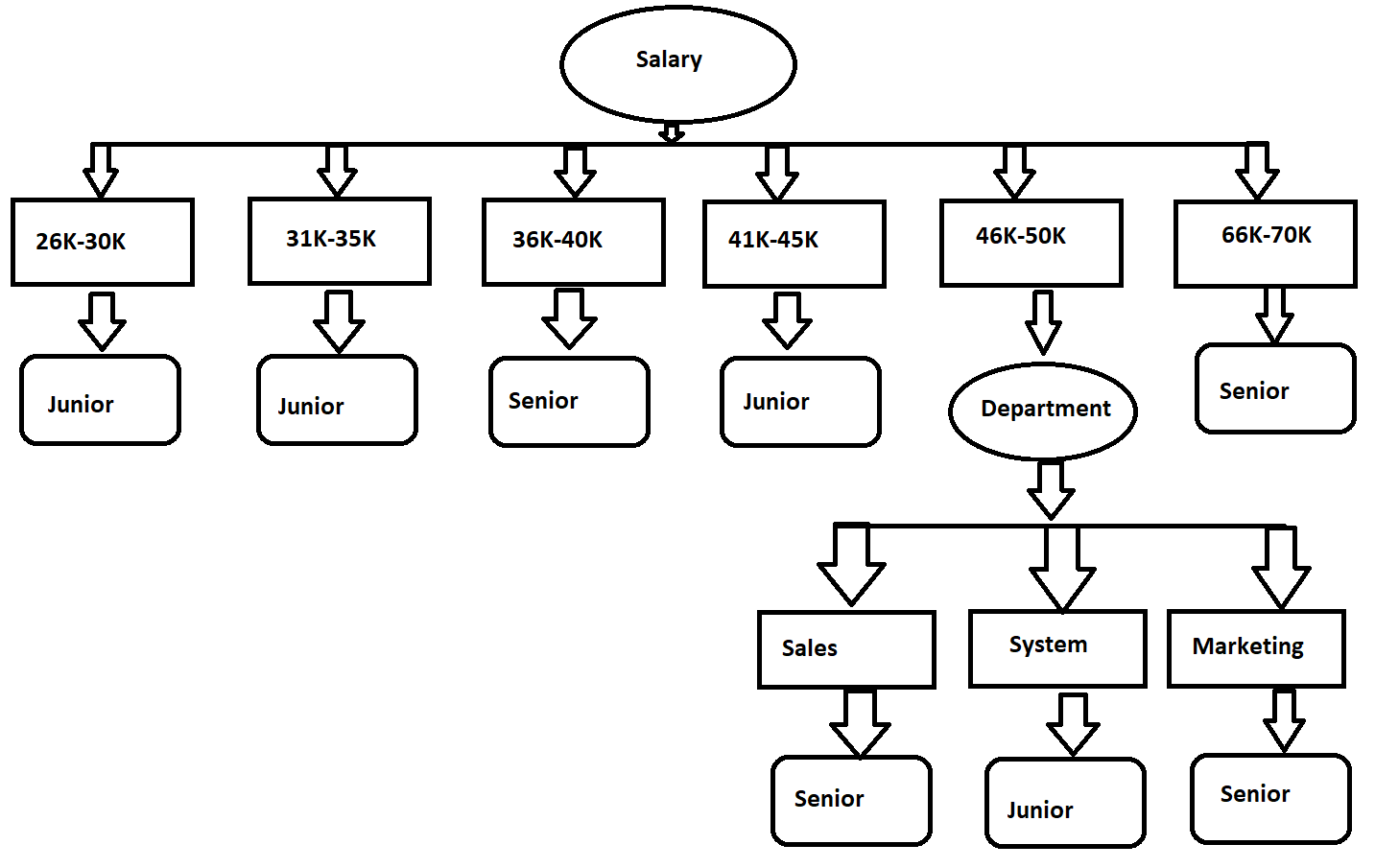
I(Pi, Ni) is calculated using = - log () - log ()

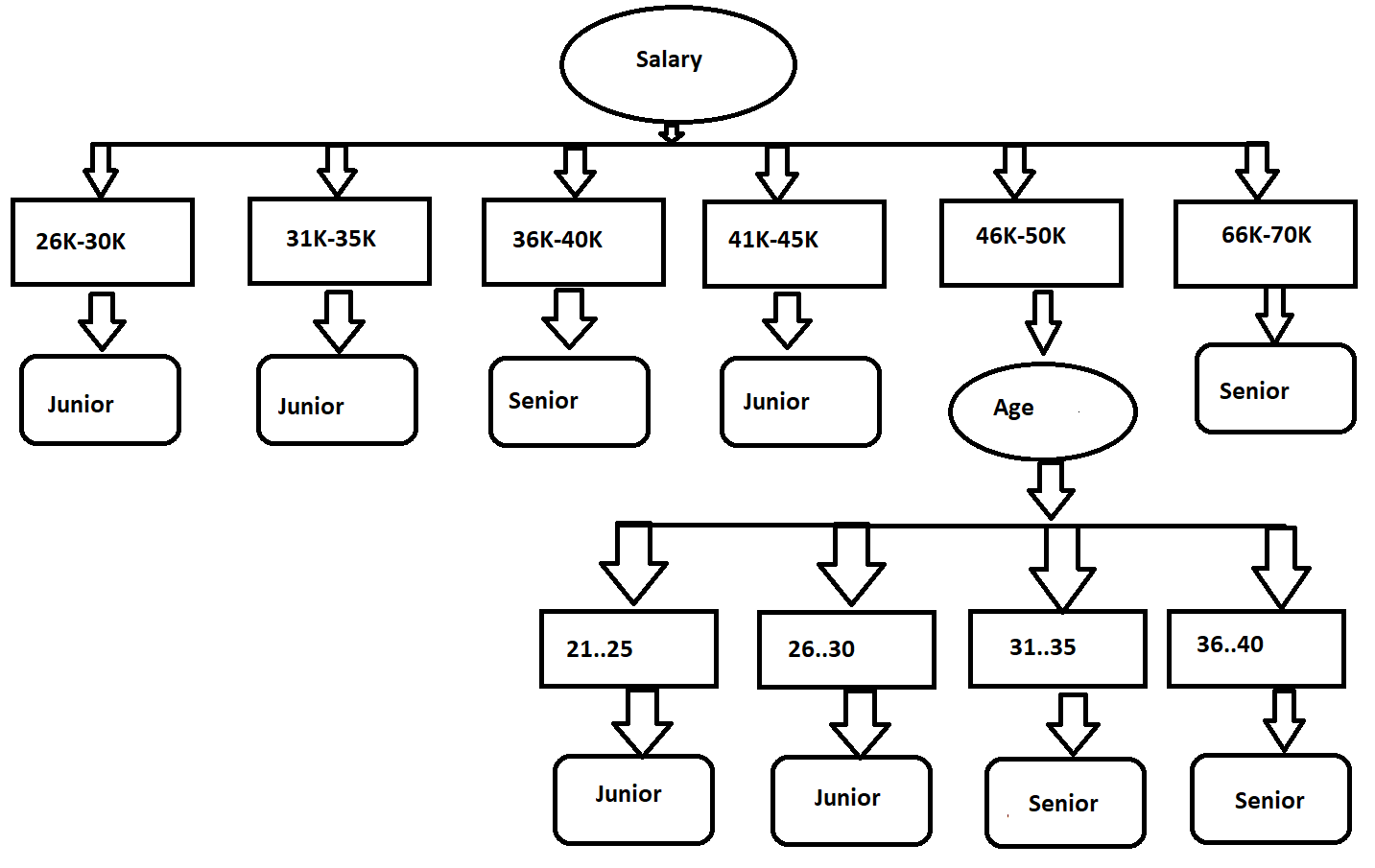
Entropy of Department =

= 0

Gain = Entropy – Entropy of Age = 0.9468-0 = 0.9468

**Gain is same for both**

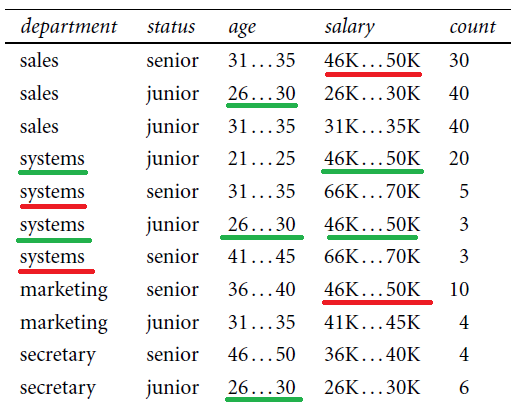




Both decision tree will work efficiently

*Only difference between considering count column and not considering count column is that if we consider count column, we get salary as root node with gain significantly more than age while if we don’t consider count, we get gain of salary and age same, in that case we can pick either as root node. Final decision tree is same for both methods.*

1. Table looks like



P(systems | junior) = 23/113

P(26..30 | junior) = 49/113

P(46-50k | junior) = 23/113

P(systems | senior) = 8/52

P(26..30 | senior) = 0 /52

P(46-50K | senior) = 40/52

P(X | junior) = P(systems | junior) x P(26..30 | junior) x P(46-50k | junior)

= 23/113 x 49/113 x 23/113

= 0.01796

P(X | senior) = P(systems | senior) x P(26..30 | senior) x P(46-50k | senior)

= 8/52 x 0/52 x 40/52

= 0.00

Tuple will be classified as junior as P(X | junior) > P(X | senior).

But, P(26..30 | senior) = 0 /52 so we have to apply laplacian correction. Increased count of 1 for each row and considering A is not systems and B is not 46-50

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department** | **Age** | **Salary** | **Count** | **Status** |
| Sales | 31..35 | 46-50 | 31 | Senior |
| Sales | 26..30 | 26-30 | 41 | Junior |
| Sales | 31..35 | 31-35 | 41 | Junior |
| Systems | 21..25 | 46-50 | 21 | Junior |
| Systems | 31..35 | 66-70 | 6 | Senior |
| Systems | 26..30 | 46-50 | 4 | Junior |
| Systems | 41..45 | 66-70 | 4 | Senior |
| Marketing | 36..40 | 46-50 | 11 | Senior |
| Marketing | 31..35 | 41-45 | 5 | Junior |
| Secretary | 46..50 | 36-40 | 5 | Senior |
| Secretary | 26..30 | 26-30 | 7 | Junior |
| A | 26..30 | B | 1 | Senior |

P(systems | junior) = 25/119

P(26..30 | junior) = 52/119

P(46-50k | junior) = 25/119

P(systems | senior) = 10/58

P(26..30 | senior) = 1 /58

P(46-50K | senior) = 42/58

P(X | junior) = P(systems | junior) x P(26..30 | junior) x P(46-50k | junior)

= 25/119 x 52/119 x 25/119

= 0.01928

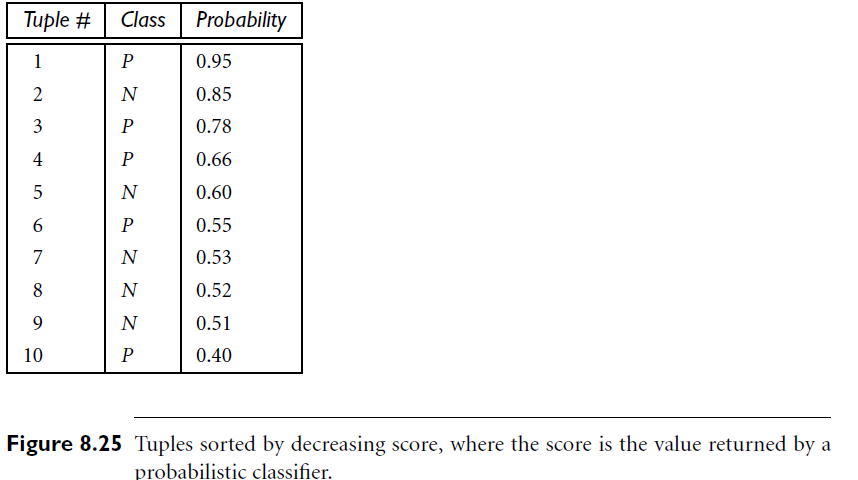
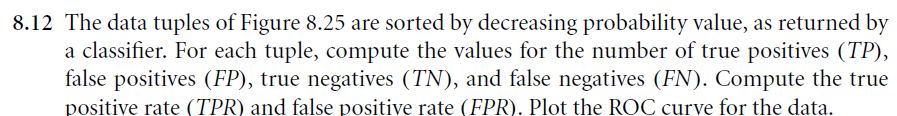
P(X | senior) = P(systems | senior) x P(26..30 | senior) x P(46-50k | senior)

= 10/58 x 1/58 x 42/58

= 0.00215

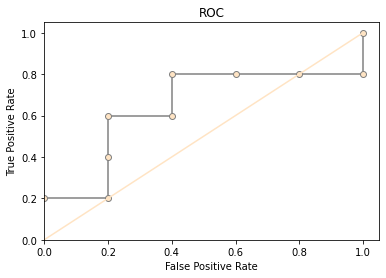
Irrespective of Laplacian correction P(X | junior) > P(X | senior)

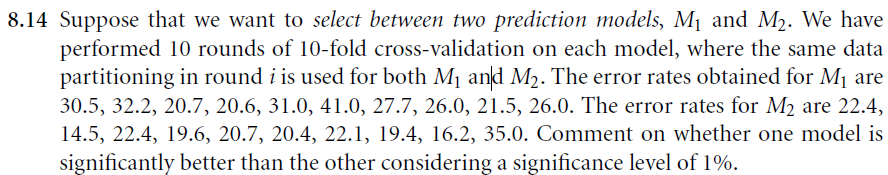
And hence, Tuple will be classified as junior.



**Answer)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tuple | Class | Probability | TP | FP | TN | FN | TPR | FPR |
| 1 | P | 0.95 | 1 | 0 | 5 | 4 | 0.2 | 0 |
| 2 | N | 0.85 | 1 | 1 | 4 | 4 | 0.2 | 0.2 |
| 3 | P | 0.78 | 2 | 1 | 4 | 3 | 0.4 | 0.2 |
| 4 | P | 0.66 | 3 | 1 | 4 | 2 | 0.6 | 0.2 |
| 5 | N | 0.6 | 3 | 2 | 3 | 2 | 0.6 | 0.4 |
| 6 | P | 0.55 | 4 | 2 | 3 | 1 | 0.8 | 0.4 |
| 7 | N | 0.53 | 4 | 3 | 2 | 1 | 0.8 | 0.6 |
| 8 | N | 0.52 | 4 | 4 | 1 | 1 | 0.8 | 0.8 |
| 9 | N | 0.51 | 4 | 5 | 0 | 1 | 0.8 | 1 |
| 10 | P | 0.4 | 5 | 5 | 0 | 0 | 1 | 1 |

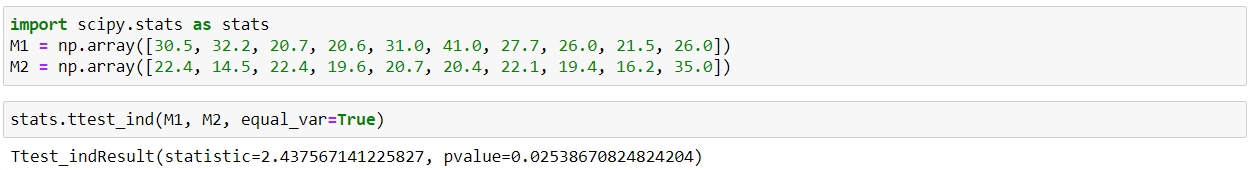




**Answer)**

Significance level = 1% = 0.01

Used scipy.stats to perform ttest



Ttest = 2.437567141225827

Pvalue=0.02538670824824204)

Degree of freedom = 9

so /2 = 0.005 and therefore t value corresponding to it is 3.25.

Range is from -3.25 to 3.25 and our ttest value lies between it.

Here pvalue that we got is greater than significance level given.

Therefore, we cannot tell which model is better as we don’t have enough evidence.