

Question 1

$P(y=0|x=A) = 0$
 $P(y=0|x=B) = 0.75$
 $P(y=0|x=C) = 1$
 $P(y=0|x=D) = 0.33$

Sort values: A D B C

(a) x in {A}, x in {A,B,D}, x in {A,D}

(b) Impurity of node: $0.5 * 0.5 = 0.25$
Impurity of x in {A,D}: $0.2 * 0.8 = 0.16$
Impurity of x in {B,C}: $0.8 * 0.2 = 0.16$
Proportion x in {A,D}: 0.5
Proportion x in {B,C}: 0.5

Impurity reduction: $0.25 - (0.5*0.16+0.5*0.16) = 0.09$

Twice this number was also correct.

Question 2

(a) Prune in t_2.
(b) $g(t_1) = (0.5 - 0.1)/(3 - 1) = 0.2$
 $g(t_3) = (0.1 - 0)/(2 - 1) = 0.1$
So $a_2 = 0.1$

(c) The resubstitution error in the root node is at most 0.5 for binary classification problems.
So the worst case total cost of the root is $0.5 + a$. The best case for a tree that is not the root node is a tree with a single split and zero resubstitution error. The cost of that tree is $0 + 2a$.
For $a = 0.5$ both trees have the same total cost, and the root node is the smallest minimizing subtree
(and of course remains the smallest minimizing subtree for $a > 0.5$)

Question 3

(a) $n(A,B,C)n(C,D)n(C,E)/n(C)^2$
(b) 12

Question 4: add(A --> C), add(D --> C)

Question 5

(a) The independence model gives a perfect fit of the observed counts, so: 32,48,8,12.
(b) Since the independence model gives a perfect fit of the observed counts, it has the same loglikelihood score as the saturated model $x \rightarrow y$. But the latter uses one more parameter to achieve that fit, at the cost of $\log(100)/2 = 2.3$.
So the difference in BIC score is -2.3.

Question 6

$|V| = 11$, Number of words in positive reviews: 12
 $P(\text{beautiful}|\text{Positive}) = (2+1)/(12+11) = 3/23$

Question 7

ACD, BDE, ABD

Question 8

(a) BC, BD, CD
(b) ABC, BC, BCD

Question 9

Authors that did not write a paper together in 2021, but did in 2022. So AD and BD.

Question 10

Let xyz denote an arbitrary frequent sequence of length $k+1$, where x is a sequence of length $k-1$ (possibly the empty sequence if $k=1$), and y and z are symbols from the set of labels. By the anti-monotonicity property of support, every subsequence of xyz is frequent. xy and xz are both length- k subsequences of xyz , and therefore frequent. The GSP candidate generation rule combines any two frequent sequences A and B of length k that match on the first $k-1$ positions to the level $k+1$ candidate $A[1:k]+B[k]$. Since xy and xz match on the first $k-1$ positions, they will be combined to the candidate xyz .