CS 422: Data Mining

Department of Computer Science
Illinois Institute of Technology
Jithin Joyson

Spring 2019: Homework 2

1 Exercises

1.1 2.

a. Gini = $1 – [(10/20)^2 + (10/20)^2] = 1 – [(1/4) + (1/4)] = 1 – 1/2 =$ **1/2**
b. For each Customer ID:
   Gini = $1 – [(1/1)^2 + (0/1)^2] = 1 – [1 + 0] = 0$
   Weighted Gini = (0 * 1/20) * 20 = **0**

c. M:
   Gini = $1 – [(6/10)^2 + (4/10)^2] = 1 – [(9/25) + (4/25)] = 1 – 13/25 = 12/25$
   F:
   Gini = $1 – [(4/10)^2 + (6/10)^2] = 1 – [(4/25) + (9/25)] = 1 – 13/25 = 12/25$
   Weighted Gini = [(1/2) * (12/25) + (1/2) * (12/25)] = **12/25**
d. Family:
   Gini = $1 – [(1/4)^2 + (3/4)^2] = 1 – [(1/16) + (9/16)] = 1 – 5/8 = 3/8$
   Sports:

   Gini = $1 – [(8/8)^2 + (0/8)^2] = 1 – [(1 + 0)] = 0$

   Luxury:

   Gini = $1 – [(1/8)^2 + (7/8)^2] = 1 – [(1/64) + (49/64)] = 1 – 50/64 = 14/64$

   Weighted Gini = [(4/20) * (3/8) + (8/20) * (0) + (8/20) * (14/64)] = [(3/40) + (7/80)] = **13/80**

e. Small:
   Gini = $1 – [(3/5)^2 + (2/5)^2] = 1 – [(9/25) + (4/25)] = 1 - 13/25 = 12/25$

   Medium:

   Gini = $1 – [(3/7)^2 + (4/7)^2] = 1 – [(9/49) + (16/49)] = 1 – 25/49 = 24/49$

   Large:

   Gini = $1 – [(2/4)^2 + (2/4)^2] = 1 – [(1/4) + (1/4)] = 1 – 1/2 = 1/2$

   Extra Large:

$\text{Gini} = 1 – [(2/4)^2 + (2/4)^2] = 1 – [(1/4) + (1/4)] = 1 – 1/2 = 1/2$

Weighted Gini = [(5/20) * (12/25) + (7/20) * (24/49) + (4/20) * (1/2) + (4/20) * (1/2)] = [3/25 + 6/35 + 1/10 + 1/10] = [6/35 + 8/25] = **172/350**

f. Since we are trying to minimize the weighed Gini values, the best attribute is **Car Type**.
g. Since Customer ID is unique, it has no prediction for the target class and does not contribute to the model.

3.

a. Entropy = -1 * [(4/9)$\log_2$(4/9) + (5/9)$\log_2$(5/9)] = **0.9910760598382222**
b. Information Gain = Initial Entropy – Final Entropy
Initial Entropy = 0. 9910760598382222

Final Entropy(a_1,T) = -1 * [(3/4)$\log_2$(3/4) + (1/4)$\log_2$(1/4)] = 0.8112781244591
Final Entropy(a_1,F) = -1 * [(1/5)$\log_2$(1/5) + (4/5)$\log_2$(4/5)] = 0.7219280948873
Final Entropy(a_1) = 4/9 * 0. 8112781244591 + 5/9 * 0.7219280948873
= 0.7616392191414825
Information Gain (a_1) = 0. 9910760598382222– 0. 7616392191414825
= **0.22943684069673975**

Final Entropy(a_2,T) = -1 * [(2/5)$\log_2$(2/5) + (3/5)$\log_2$(3/5)] = 0.9709505944546
Final Entropy(a_2,F) = -1 * [(2/4)$\log_2$(2/4) + (2/4)$\log_2$(2/4)] = 1.0
Final Entropy(a_2) = 5/9 * 0.9709505944546 + 4/9 * 1.0
= 0.9838614413637048
Information Gain (a_2) = 0. 9910760598382222– 0.9838614413637048
= **0.007214618474517431**

c. Information Gain = Initial Entropy – Final Entropy
Initial Entropy = 0. 9910760598382222

Final Entropy(a_3,< 2.0) = -1 * [(1/1)$\log_2$(1/1) + (0/1)$\log_2$(0/1)] = 0.0
Final Entropy(a_3,> 2.0) = -1 * [(3/8)$\log_2$(3/8) + (5/8)$\log_2$(5/8)]
= 0.954434002924965
Final Entropy(a_3, 2.0) = 1/9 * 0.0 + 8/9 * 0.954434002924965
= 0.8483857803777467
Information Gain (a_1, 2.0) = 0. 9910760598382222– 0. 7616392191414825
= **0.14269027946047552**

Final Entropy(a_3,< 3.5) = -1 * [(1/2)$\log_2$(1/2) + (1/2)$\log_2$(1/2)] = 1.0
Final Entropy(a_3,> 3.5) = -1 * [(3/7)$\log_2$(3/7) + (4/7)$\log_2$(4/7)]

= 0.9852281360342515

Final Entropy(a_3, 3.5) = 2/9 * 0. 8112781244591 + 7/9 * 0.7219280948873

= 0.9885107724710845

Information Gain (a_1, 3.5) = 0. 9910760598382222– 0. 7616392191414825

= **0.002565287367137681**

Final Entropy(a_3,< 4.5) = -1 * [(2/3)$\log_2$(2/3) + (1/3)$\log_2$(1/3)]

= 0.9182958340544896

Final Entropy(a_3,> 4.5) = -1 * [(2/6)$\log_2$(2/6) + (4/6)$\log_2$(4/6)]

= 0.9182958340544896

Final Entropy(a_3, 4.5) = 3/9 * 0. 8112781244591 + 6/9 * 0.7219280948873

= 0.9182958340544896

Information Gain (a_1, 4.5) = 0. 9910760598382222– 0. 7616392191414825

= **0.07278022578373267**

Final Entropy(a_3,< 5.5) = -1 * [(2/4)$\log_2$(2/4) + (2/4)$\log_2$(2/4)] = 1.0

Final Entropy(a_3,> 5.5) = -1 * [(2/5)$\log_2$(2/5) + (3/5)$\log_2$(3/5)]

= 0.9709505944546686

Final Entropy(a_3, 5.5) = 4/9 * 0. 8112781244591 + 5/9 * 0.7219280948873

= 0.9838614413637048

Information Gain (a_1, 5.5) = 0. 9910760598382222– 0. 7616392191414825

= **0.007214618474517431**

Final Entropy(a_3,< 5.5) = -1 * [(2/5)$\log_2$(2/5) + (3/5)$\log_2$(3/5)]

= 0.9709505944546686

Final Entropy(a_3,> 5.5) = -1 * [(2/4)$\log_2$(2/4) + (2/4)$\log_2$(2/5)] = 1.0

Final Entropy(a_3, 5.5) = 5/9 * 0. 8112781244591 + 4/9 * 0.7219280948873

= 0.9838614413637048

Information Gain (a_1, 5.5) = 0. 9910760598382222– 0. 7616392191414825

= **0.007214618474517431**

Final Entropy(a_3,< 6.5) = -1 * [(3/6)$\log_2$(3/6) + (3/6)$\log_2$(3/6)] = 1.0

Final Entropy(a_3,> 6.5) = -1 * [(1/3)$\log_2$(1/3) + (2/3)$\log_2$(2/3)]

= 0.9182958340544896

Final Entropy(a_3, 6.5) = 6/9 * 0. 8112781244591 + 3/9 * 0.7219280948873

= 0.9727652780181631

Information Gain (a_1, 6.5) = 0. 9910760598382222– 0. 7616392191414825

= **0.018310781820059074**

Final Entropy(a_3,< 7.5) = -1 * [(4/7)$\log_2$(4/7) + (3/7)$\log_2$(3/7)]

= 0.9852281360342515

Final Entropy(a_3,> 7.5) = -1 * [(0/2)$\log_2$(0/2) + (2/2)$\log_2$(2/2)] = 0.0
Final Entropy(a_3, 7.5) = 7/9 * 0. 8112781244591 + 2/9 * 0.7219280948873
= 0.7662885502488623
Information Gain (a_1, 7.5) = 0. 9910760598382222– 0. 7616392191414825
= **0.018310781820059074**

Final Entropy(a_3,< 7.5) = -1 * [(4/8)$\log_2$(4/8) + (4/8)$\log_2$(4/8)] = 1.0
Final Entropy(a_3,> 7.5) = -1 * [(0/1)$\log_2$(0/1) + (1/1)$\log_2$(1/1)] = 0.0
Final Entropy(a_3, 7.5) = 8/9 * 0. 8112781244591 + 1/9 * 0.7219280948873
= 0.8888888888888888
Information Gain (a_1, 7.5) = 0. 9910760598382222– 0. 7616392191414825
= **0.10218717094933338**

d. Since we are trying to maximize information gain, $\mathbf{a_1}$ would be the best split.
e. Misclassification(a_1) = min((9 – 7)/9,(9 – 2)/9) = min(2/9, 7/9) = 2/9
Misclassification(a_1) = min((9 – 4)/9,(9 – 5)/9) = min(5/9, 4/9) = 4/9
Since we are trying to minimize misclassification error, $\mathbf{a_1}$ would be the best split.
f. $A_{1:}$
True : 1 – [$(3/4)^2$ + $(1/4)^2$] = 1 – [(9/16) + (1/16)] = 1 - 5/8 = 3/8
False: 1 = [$(1/5)^2$ + $(4/5)^2$] = 1 – [(1/25) + (16/25)] = 1 – 17/25 = 8/25
Weighted Gini = 4/9(3/8) + 5/9(8/25) = 1/6 + 8/45 = **93/270**
$A_{2:}$
True : 1 – [$(2/5)^2$ + $(3/5)^2$] = 1 – [(4/25) + (9/24)] = 1 - 13/25 = 12/25
False: 1 = [$(2/4)^2$ + $(2/4)^2$] = 1 – [(1/4) + (1/4)] = 1 – 1/2 = 1/2
Weighted Gini = 5/9(12/25) + 4/9(1/2) = 4/15 + 2/9 = **66/135**
Since we are trying to minimize the weighed Gini values, the best split is on $\mathbf{a_1}$.

5.

a. Information Gain = Initial Entropy – Final Entropy
Initial Entropy = -1 * [(4/10)$\log_2$(4/10) + (6/10)$\log_2$(6/10)]
= 0.9709505944546686

Final Entropy (A, T) = -1 * [(4/7)$\log_2$(4/7) + (3/7)$\log_2$(3/7)]
= 0.9852281360342515
Final Entropy (A, F) = -1 * [(0/3)$\log_2$(0/3) + (3/3)$\log_2$(3/3)] = 0.0
Final Entropy (A) = 7/10 * 0.0 + 3/10 * 0.954434002924965
= 0.6896596952239761
Information Gain (A) = 0. 9910760598382222– 0. 7616392191414825

= **0.2812908992306925**

Final Entropy (B, T) = -1 * [(3/4)$\log_2$(3/4) + (1/4)$\log_2$(1/4)]
= 0.8112781244591328
Final Entropy (B, F) = -1 * [(1/6)$\log_2$(1/6) + (5/6)$\log_2$(5/6)]
= 0.6500224216483541
Final Entropy (B) = 4/10 * 0.0 + 6/10 * 0.954434002924965
= 0.7145247027726656
Information Gain (B) = 0. 9910760598382222– 0. 7616392191414825
= **0.256425891682003**

Since we are trying to maximize information gain, **A** would be the best split.

b. Gain = Initial Gini – Final Gini
Initial Gini = 1 – [$(4/10)^2$ + $(6/10)^2$] = 1 – [(16/100 + 36/100)] = 48/100 = 12/25
A
True: 1 – [$(4/7)^2$ + $(3/7)^2$] = 1 – [(16/49) + (9/49)] = 1 - 25/49 = 24/49
False: 1 = [$(0/3)^2$ + $(3/3)^2$] = 1 – [(0/9) + (9/9)] = 1 – 9/9 = 0
Weighted Gini = 7/10(24/49) + 3/10(0) = 12/35 = 12/35

Gain = 12/25 – 12/35 = **24/175**
B
True: 1 – [$(3/4)^2$ + $(1/4)^2$] = 1 – [(9/16) + (1/16)] = 1 – 5/8 = 3/8
False: 1 = [$(1/6)^2$ + $(5/6)^2$] = 1 – [(1/36) + (25/36)] = 1 – 13/18 = 5/18
Weighted Gini = 4/10(3/8) + 6/10(5/18) = 3/20 + 1/6 = 19/60

Gain = 12/25 – 19/60= **49/300**

Since we are trying to maximize gain, the best split is on **B**.

c. As illustrated by parts (a) and (b), it is still possible for gain in Gini index and information gain favor different attributes since the entropy and Gini values can vary throughout attributes; an attribute can have similar distributions respective to their own entropy and Gini values but not with respect to other attribute's entropy and Gini values.

1.2 18. Assume there are 10 records in the test dataset. N =10

a. Actual Yes = P(Yes)*N = 1/2 * 10 = 5

Actual No = N – Actual Yes = 10 – 5 = 5
Accuracy = (TP + TN)/ N
TP = Actual Yes * P(Yes of classifier) = 5 * 1 = 5
TN = Actual No * P(No of classifier) = 5 * 0 = 0
Accuracy = (5 + 0)/10 = 1/2
Error = 1 – Accuracy = 1 – 1/2 = 1/2 = **50%**

b. Actual Yes = P(Yes)*N = 1/2 * 10 = 5
Actual No = N – Actual Yes = 10 – 5 = 5
Accuracy = (TP + TN)/ N
TP = Actual Yes * P(Yes of classifier) = 5 * .8 = 4
TN = Actual No * P(No of classifier) = 5 * .2 = 1
Accuracy = (4 + 1)/10 = 1/2
Error = 1 – Accuracy = 1 – 1/2 = 1/2 = **50%**

c. Actual Yes = P(Yes)*N = 2/3 * 10 = 20/3
Actual No = N – Actual Yes = 10 – 20/3 = 10/3
Accuracy = (TP + TN)/ N
TP = Actual Yes * P(Yes of classifier) = 20/3 * 1 = 20/3
TN = Actual No * P(No of classifier) = 10/3 * 0 = 0
Accuracy = (20/3 + 0)/10 = 20/30 = 2/3
Error = 1 – Accuracy = 1 – 2/3 = 1/3 = **33.3%**

d. Actual Yes = P(Yes)*N = 2/3 * 10 = 20/3
Actual No = N – Actual Yes = 10 – 20/3 = 10/3
Accuracy = (TP + TN)/ N
TP = Actual Yes * P(Yes of classifier) = 20/3 * 2/3 = 40/9
TN = Actual No * P(No of classifier) = 10/3 * 1/3 = 10/9
Accuracy = (40/9 + 10/9)/10 = 50/90 = 5/9
Error = 1 – Accuracy = 1 – 5/9 = 4/9 = **44.4%**

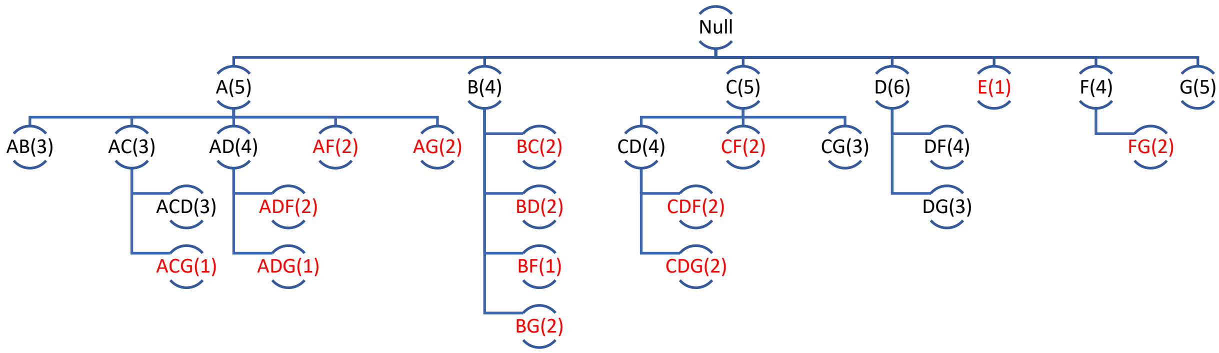
1.3 1. Using Database 8.2

a. Binary Database:

	D	A	B	C	D	E	F	G
	1	1	1	1	1	0	0	0
	2	1	0	1	1	0	1	0
	3	1	0	1	1	1	0	1
	4	1	1	0	1	0	1	0
	5	0	1	1	0	0	0	1
	6	0	0	0	1	0	1	1

	7	1	1	0	0	0	0	1
	8	0	0	1	1	0	1	1

Minsup = 3



Final Frequent Items:

{D (6), C (5), G (5), A (5), F (4), B (4), AC (3), ACD (3), AD (4), CD (4), CG (3), DF (4), DG (3)}

b. Minsup = 2

Frequency Table:

A	B	C	D	E	F	G
5	4	5	6	1	4	5

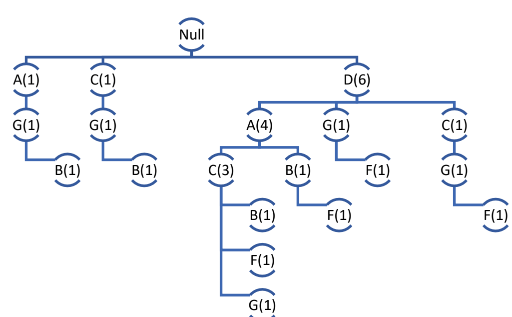
Sort by Frequency:

D	A	C	G	B	F
6	5	5	5	4	4

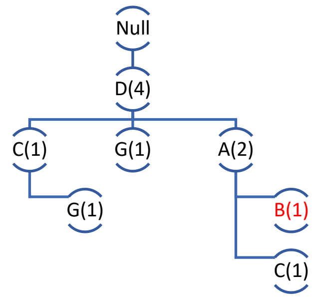
Transaction arranged by frequency:

	ABCD	DACB
	ACDF	DACF
	ACDEG	DACGE
	ABDF	DABF
	BCG	CGB
	DFG	DGF
	ABG	AGB
	CDFG	DCGF

FP – Growth Tree: R:



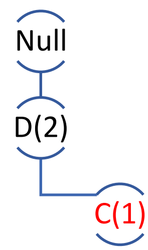
$\mathbf{R_F}$: DCGF cnt(F) = 1
DGF cnt(F) = 1
DABF cnt(F) = 1
DACF cnt(F) = 1



Sup(D) = 4
Sup(C) = 1 + 1 = 2
Sup(G) = 1 + 1 = 2
Sup(A) = 2
Sup(B) = 1

Pruning B does not create Linear path so keep projecting:

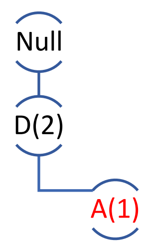
$(R_F)_G$: DG cnt(G) = 1
DCG cnt(G) = 1



Sup(D) = 2
Sup(C) = 1

Frequent Items: Sup(FG) = 2
Sup(FGD) = 2

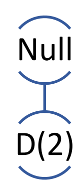
$(R_F)_C$: DC cnt(C) = 1
DAC cnt(C) = 1



Sup(D) = 2
Sup(A) = 1

Frequent Items: Sup(FC) = 2
Sup(FCD) = 2

$(R_F)_A$: DA cnt(A) = 2



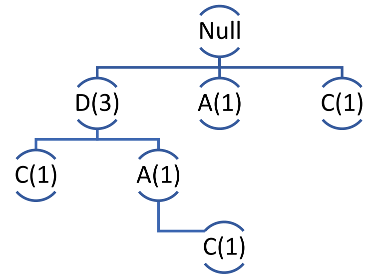
Sup(D) = 2

Frequent Items: Sup(FA) = 2
Sup(FAD) = 2

$(R_F)_D$: D cnt(D) = 4

Frequent Items: Sup(FD) = 4

R_G: AG cnt(G) = 1
CG cnt(G) = 1
DG cnt(G) = 1
DCG cnt(G) = 1
DACG cnt(G) = 1



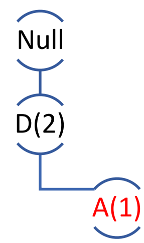
Sup(D) = 3
Sup(A) = 1 + 1 = 2
Sup(C) = 1 + 1 + 1 = 3

Pruning does not create Linear path so keep projecting:

$(R_G)_D$: D cnt(D) = 3

Frequent Items: Sup(GD) = 3

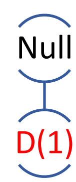
$(R_G)_C$: C cnt(C) = 1
DC cnt(C) = 1
DAC cnt(C) = 1



Sup(D) = 2
Sup(A) = 1

Frequent Items: Sup(GC) = 3
Sup(GCD) = 2

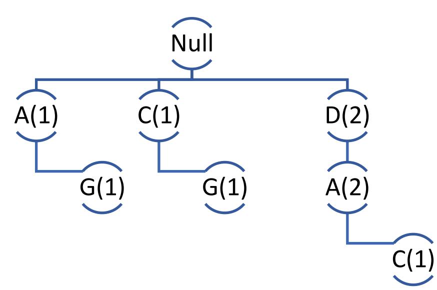
$(R_G)_A$: A cnt(A) = 1
DA cnt(A) = 1



Sup(D) = 1

Frequent Items: Sup(GA) = 2

R_B: AGB cnt(B) = 1
CGB cnt(B) = 1
DACB cnt(B) = 1
DAB cnt(B) = 1



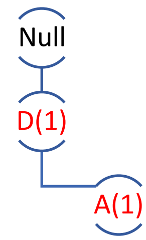
Sup(D) = 2
Sup(A) = 1 + 2 = 3
Sup(C) = 1 + 1 = 2
Sup(G) = 1 + 1 = 2

Pruning does not create Linear path so keep projecting:

$(R_B)_D$: D cnt(D) = 2

Frequent Items: Sup(BD) = 2

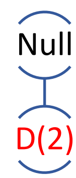
$(R_B)_C$: C cnt(C) = 1
DAC cnt(C) = 1



Sup(D) = 1
Sup(A) = 1

Frequent Items: Sup(BC) = 2

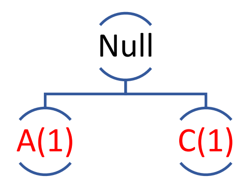
$(R_B)_A$: A cnt(A) = 1
DA cnt(A) = 2



Sup(D) = 2

Frequent Items: Sup(BA) = 3
Sup(BAD) = 2

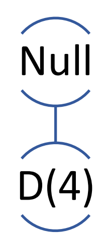
$(R_B)_G$: AG cnt(G) = 1
CG cnt(G) = 1



Sup(A) = 1
Sup(C) = 1

Frequent Items: Sup(BG) = 2

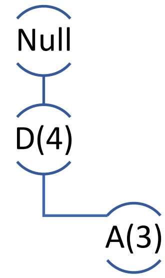
R_A: A cnt(A) = 1
DA cnt(A) = 4



Sup(D) = 4
Sup(A) = 1 + 4 = 5

Linear so:
Frequent Items: Sup(AD)= 4

R_C: C cnt(C) = 1
DAC cnt(C) = 3
DC cnt(C) = 1



Sup(D) = 3
Sup(A) = 3
Sup(C) = 1 + 3 + 1 = 5

Linear so:
Frequent Items: Sup(CDA) = 3
Sup(CD) = 4
Sup(CA) = 3

R_D: D cnt(D) = 6

Sup(D) = 6

Linear so:
Frequent Items: {}

Skip R_E since small minsup

Final Frequent Items:
{FG (2), FGD (2), FC (2), FCD (2), FA (2), FAD (2), FD (4), GD (3), GC (3), GCD (2), GA (2), BD (2), BC (2), BA (3), BAD (2), BG (2), AD (4), CDA (3), CD (4), CA (3), D (6), C(5), G (5), A (5), F (4), B (4)}

2. No minsup:

	Rule	Confidence
	{AB} => {E}	66 %
	{AE} => {B}	100 %
	{BE} => {A}	50 %
	{E} => {AB}	50 %
	{A} => {BE}	50 %
	{B} => {AE}	40 %
	{} => A,B,E	100 %
	A,B,E => {}	0 %

6. a. Since there are 11 leaves and the leaves are simple items, the search space should be 11 items. But the taxology forces the higher nodes to be accessed so these are included in the search space (11 + 4) = 15.
b. Since the replacement with the parent only happens when it exists, a frequency set of all high-level nodes won't be replaced and the support will

be equal to the children's support. In any other scenario, more single-level nodes get replaced by high-level nodes and the support thus increases.

Therefore, the new itemset X' will **(iv)** more than or equal to support of X.

1.4 Binary Confusion Matrices:

Setosa	Actual		
Prediction		True	False
	True	TP = 8	FP = 0
	False	FN = 0	TN = (10+9+1+2) = 22

Sensitivity: TP/(TP + FN) = 8/8 = 1.00
Specificity: TN/(TN + FP) = 22/22 = 1.00
Precision: TP/(TP + FP) = 8/8 = 1.00

Versicolor	Actual		
Prediction		True	False
	True	TP = 10	FP = 1
	False	FN = 2	TN = (8 + 9) = 17

Sensitivity: TP/(TP + FN) = 10/12 = 0.83
Specificity: TN/(TN + FP) = 17/18 = 0.94
Precision: TP/(TP + FP) = 10/11 = 0.91

Virginica	Actual		
Prediction		True	False
	True	TP = 9	FP = 2
	False	FN = 1	TN = (10+8) = 18

Sensitivity: TP/(TP + FN) = 9/10 = 0.90
Specificity: TN/(TN + FP) = 18/20 = 0.90
Precision: TP/(TP + FP) = 9/11 = 0.82