Data Mining 2021 midterm exam. (11/22/2021)

1. (20%) (Regression) The following dataset is a simplified cereal dataset. Please build a regression model using “RAING” as the response variable and “SUGARS” as the predictor variable (15%). Then, find the R-square for this linear regression problem. (5%)

Brand	SUGARS	RATING
1	6	56
2	8	48
3	5	60
4	0	80
5	8	50
6	10	40
7	14	24
8	8	46

Related equations:

Mean(Y) =50.5 Mean(X)=7.375

b1=Sxy/Sxx；b0=mean(y)-b1*mean(x)；

$R^2=S^2xy/(Sxx*Syy)=SSR/SST=(SST-SSE)/SST$

S_{xy}=cov(X,Y); S_{xx}=Varance(X)

Sol: output from lm of R

regressed model

Call:

lm(formula = y ~ x)

Coefficients:

(Intercept) x

80 -4

predicted values=

1 2 3 4 5 6 7 8

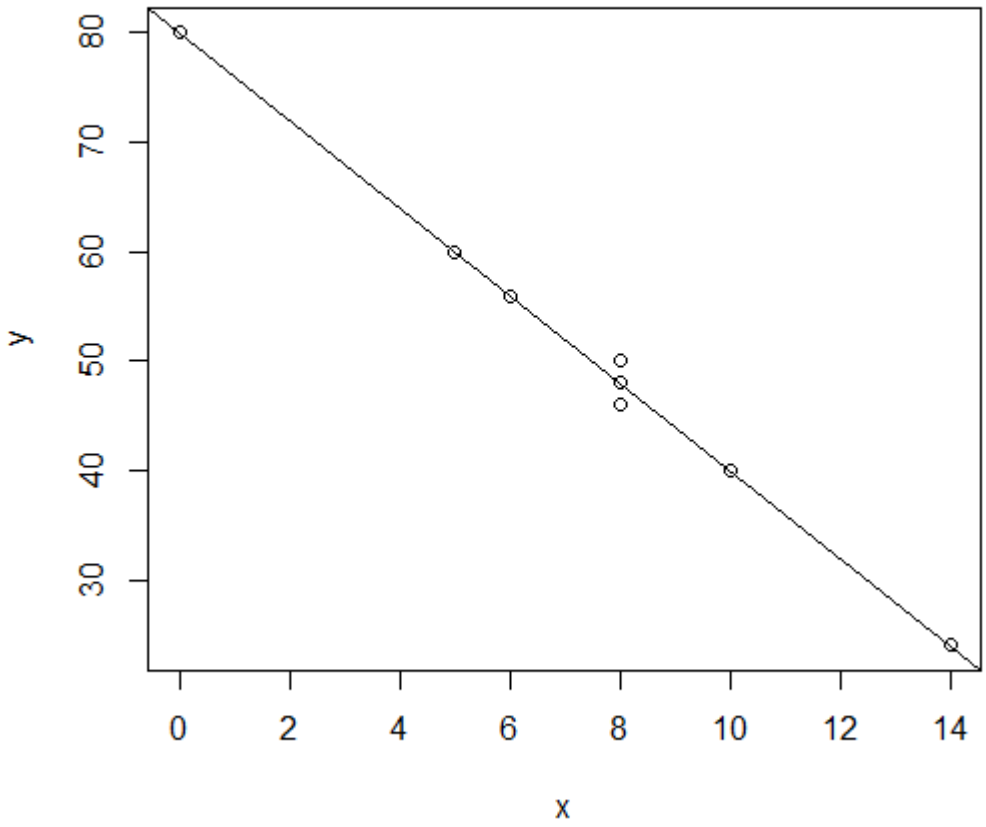
56 48 60 80 48 40 24 48

SSE= 8

SST= 1830

R-Square = 0.9956284

Regressed Fig. :



2. (20%) (Association rules) Given a transaction database in the following.

Answer the following questions.

Transaction ID	Items Bought
T100	{a,b,c,d}
T200	{a,c,e}
T300	{a,d,e}
T400	{a,b,c,d}
T500	{a,b,c}
T600	{b,c,d}
T700	{a,b}

a. Given ***min_sup*** = 40%, use the "*Apriori*" algorithm to find all of the frequent itemsets. (10%).

b. Compute the confidence and lift for rule "ab→c". (10%)

Min_sup= 0.4, min. support count= 0.4 * 7 = 2.8 , must be greater than or equal to 3 to make an itemset frequent.

Question a:

a: 6, b:5, c:5, d:4, ~~e:2~~; ab: 4, ac:4, ad:3, bc:4, bd:3, cd:3; abc:3, ~~abd:2~~, ~~acd:2~~,bcd:3; <- frequent itemsets

Question b:

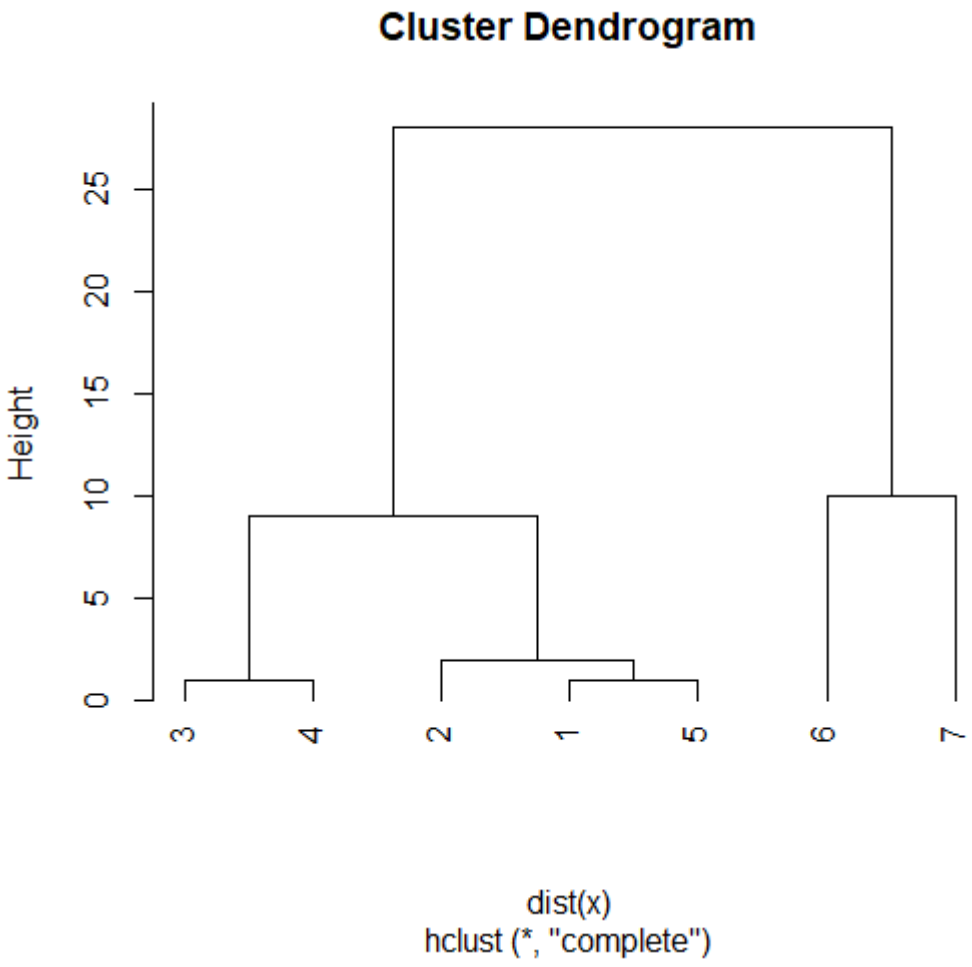
Conf. (ab->c)=# of trans. Containing {a,b,c}/ # of trans. Containing {a,b} = 3/4= 75%

Lift(ab->c) = prob.({a,b,c})/(P({a,b}) * P({c}))

(3/7)/(4/7 *5/7) =21/20 =1.05

3. (20%) (Clustering) Given a set of one-dimensional data points of {2,4,10, 11,3,20,30}
   a. Partition the data points into 2 clusters using "COMPLETE link" for computing the distance between two clusters. Draw the dendrogram. (10%)
   b. Use the K-means method to cluster the data points into two clusters with the two initial mean points of m1=2 and m2=11. (10%)

Sol: a.



b.

K-means clustering with 2 clusters of sizes 5, 2

Cluster means:

	number
1	6
2	25

Clustering vector:

[1] 1 1 1 1 1 2 2

Within cluster sum of squares by cluster:

```
[1] 70 50
 (between_SS / total_SS =  81.1 %)

Available components:

[1] "cluster"      "centers"      "totss"
[4] "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

4. (20%) (Decision Trees) The following is the dataset for problem 4.

Customer ID	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C1
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C0

(a) Compute the Gini index for the Gender attribute. (5%)

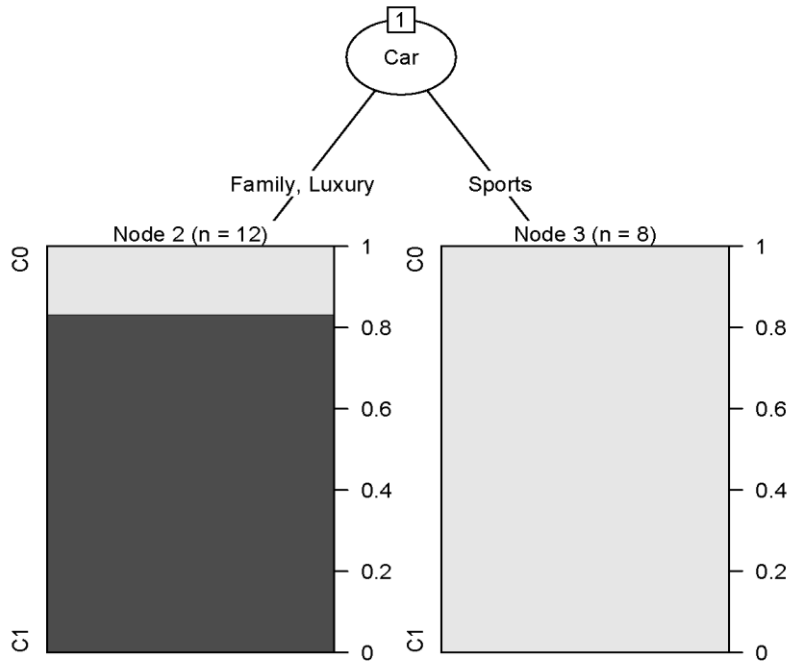
(b) Compute the Gini index for the Car type attribute using multiway split. (5%)

(c) Compute the Gini index for the Shirt Size attribute using a multiway split. (5%)

(d) Build a two-level decision tree and calculate the classification accuracy of the tree based on the training examples. (5%)

Note: a multiway split can split a node into more than two child nodes. In other words, the Shirt Size attribute divides a node into four child nodes.

Multiway-Split decision by CHAID



Output:

Real class

[1] C1 C0 C0 C0 C0 C0 C0 C0 C0 C0 C1 C1 C1 C1 C1 C1 C1 C1 C1 C0

Levels: C0 < C1

Predicted

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

C1 C0 C0 C0 C0 C0 C0 C0 C0 C1 C1 C1 C1 C1 C1 C1 C1 C1 C1

Levels: C0 < C1

Confusion Matrix

	Ypred	
	C0	C1
C0	8	2
C1	0	10

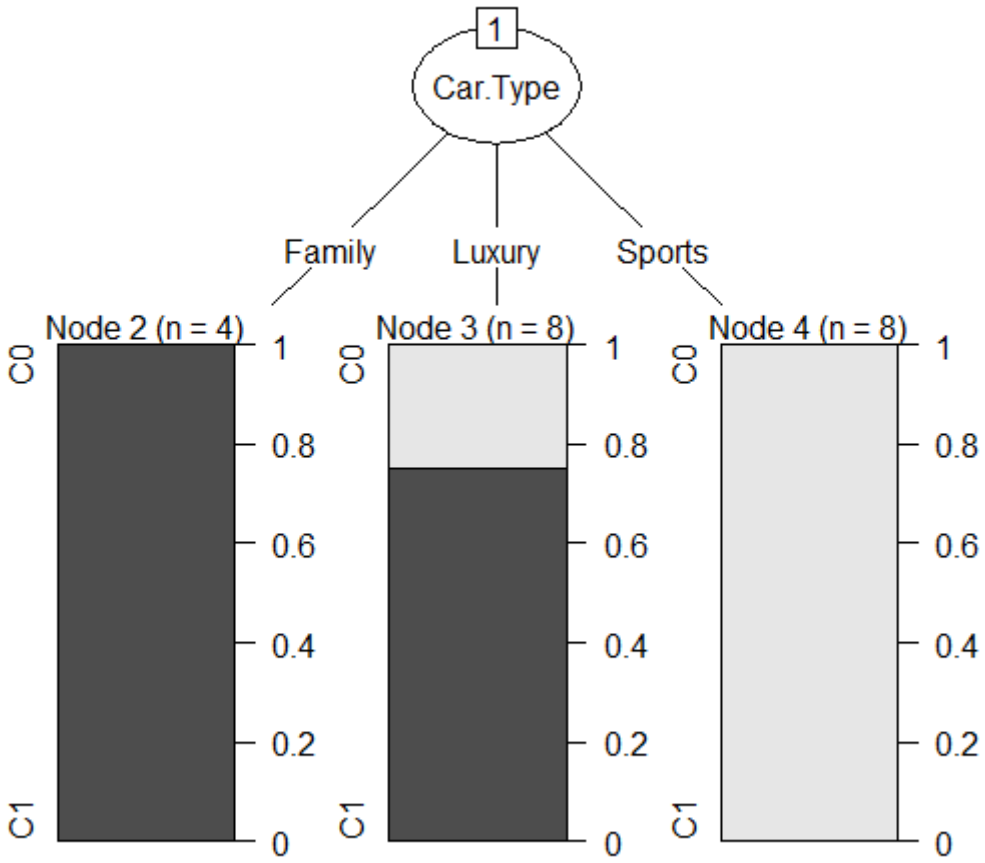
Accuracy=

[1] 0.9

Multiway split by J48 (ID3)

```
> summary(resultJ48)

=== Summary ===

Correctly Classified Instances          18               90
   %
Incorrectly Classified Instances        2               10
   %
Kappa statistic                          0.8
Mean absolute error                      0.15
Root mean squared error                  0.2739
Relative absolute error                 30      %
Root relative squared error             54.7723 %
Total Number of Instances               20

=== Confusion Matrix ===

  a  b   <-- classified as
  8  2 |  a = C0
  0 10 |  b = C1
> plot(resultJ48)
```

The Tree:



5. (20%) (Naïve Bayes classifier) Based on the following dataset, predict the Class label of the sample of X(A=0, B=0, C= 1, D=2). Note that D can have three different values 0, 1, and 2. The dataset does not contain any samples with D equal to 2.

ID	A	B	C	D	Class
1	0	0	1	0	N
2	1	0	1	0	P
3	0	1	0	1	N
4	1	0	0	1	N
5	1	0	1	0	P
6	0	0	1	0	P
7	1	1	0	1	N
8	0	0	0	1	N
9	0	1	0	0	P

10 1 1 1 0 P

Ans.

Theorically, P(Class =”P”) = P(Class=”N”)=1/2
P(Class =”P”|X)= P(X|Class= “P”) * P(Class=”P”)/P(X)
P(Class =”N”|X)= P(X|Class= “N”) * P(Class=”N”)/P(X)
Since P(X) is a constant, and both prior probabilities are equal, we only have to compare the tqo formulas P(X|Class= “P”) and P(X|Class= “N”).

P(X|Class=”P”)=P(A=0|Class=”P”)*P(B=0|Class=”P”)*P(C=1|Class=”P”)*P(D=2|Class=”P”)
=(2+1)/(5+2) * (3+1)/(5+2) * (4+1)/(5+2)*(0+1)/(5+3) (1)
P(X|Class=”N”)=P(A=0|Class=”N”)*P(B=0|Class=”N”)*P(C=1|Class=”N”)*P(D=2|Class=”N”)
=(3+1)/(5+2) * (3+1)/(5+2) * (1+1)/(5+2)*(0+1)/(5+3) (2)

Since (1) is larger than (2), X sould be calssified as Class label “P”.

However, the R output shows diffenent result!

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = comp1[, -5], y = comp1[, 5], laplace = 1,
importance = TRUE)

A-priori probabilities:
comp1[, 5]
N P
0.5 0.5

Conditional probabilities:
A
comp1[, 5] [,1] [,2]
N 0.4 0.5477226
P 0.6 0.5477226

B

comp1[, 5] [,1] [,2]
N 0.4 0.5477226
P 0.4 0.5477226

C
comp1[, 5] [,1] [,2]
N 0.2 0.4472136
P 0.8 0.4472136

D
comp1[, 5] [,1] [,2]
N 0.8 0.4472136
P 0.0 0.0000000

[1] P P N N P P N N P P
Levels: N P
Confusion Matrix
bayes.pred
N P
N 4 1
P 0 5
Accuracy:
[1] 0.9

Predicting the test sample[1] N← this is wrong!

Levels: N P

However, if I add two samples, (1,1,0,2,N) and (1,1,0,2,P) in the dataset. The result is as the following.

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = comp1[, -5], y = comp1[, 5], laplace = 1,
importance = TRUE)

A-priori probabilities:

```
comp1[, 5]
   N      P
0.5 0.5

Conditional probabilities:
                A
comp1[, 5]          [,1]          [,2]
            N 0.5000000 0.5477226
            P 0.6666667 0.5163978

                B
comp1[, 5] [,1]          [,2]
            N    0.5 0.5477226
            P    0.5 0.5477226

                C
comp1[, 5]          [,1]          [,2]
            N 0.1666667 0.4082483
            P 0.6666667 0.5163978

                D
comp1[, 5]          [,1]          [,2]
            N 1.0000000 0.6324555
            P 0.3333333 0.8164966

 [1] P P N N P P N N N P N N
Levels: N P
Confusion Matrix
     bayes.pred
       N P
   N 5 1
   P 2 4
Accuracy:
[1] 0.75
```

Predicting the test sample[1] P This is correct!

```
Levels: N P
```

The problem is that value of D=2 needs to be involved in the training (or fitting) process; otherwise, it will give you a wrong answer.

I also checked with the Python version. It directly complained that value of D=2 is not recognized, and refused to continue!

SO, everyone get 20 points on this question. Whoever correctly formulated the equation and had the correct answer will get 10 extra points! Sometimes, you cannot trust the textbook! (Jiawei Han's ppt)