Data Mining HW-3
Ehsan Mosadegh

Q.1

A- rule accuracy
R1= 9/9= 1
R2= 65/(65+40)= 0.61
Based on rule accuracy measure R1 is the better rule.

B- ratio statistics
R1= 2[(9log2 (9/ (70/270))) + (0)] = 92.14
R2= 2[(65log2 (65/ (70/270))) + 40log2 (40/ (200/270))] = 1496.6
Based on ratio statistics measure R2 is the better rule.

C- laplas measure
R1= (9+1/9+2)= 0.9
R2= (65+1/(65+40+2))= 66/107= 0.61
Based on laplace measure, R1 is the better rule.

D- m-estimate measure
R1= 9+(2*0.26)/(9+2)= 0.86
R2= 65+(2*0.26)/(105+2)= 0.61
Based on m-estimate measure, R1 is the better rule.

Q.2
Naive bayes classifiers

A- estimate conditional probability
P(A=1|+)= 5/7
P(B=1|+)= 4/7
P(C=1|+)= 1/7
P(A=1|-)= 1/3
P(B=1|-)= 1/3
P(C=1|-)= 1

B- predict class label for test sample= (A=0, B=1, C=1)
We first calculate the probability of a sample belongs to each negative and positive class based on naive bayes assumption (eq. 5.15), and then we select the higher/maximum probability between 2 classes.
$P(C|X)= P(C)\ \pi P(Xi|C)/\ P(X)$
max{P(+|X) , P(-|X)}
P(A=0|+)= 2/7, P(A=0|-)= 2/3

P(+|X)= P(+)P(A=0|+)P(B=1|+)P(C=1|+)/ P(X)= 0.016
P(-|X)= P(-)P(A=0|-)P(B=1|-)P(C=1|-)/ P(X)= 0.066

max{P(+|X) , P(-|X)} = negative class = 0.066

C- time complexity of training Naive Bayes classifier …

For training, Naive Bayes Classifier calculates the posterior probability of every feature value given each class. The time complexity is O(C*N*F), where C is the number of classes, N is the number of training samples, and F is the number of features in a sample.

D- testing time complexity of Naive Bayes classifier ...

For testing, Naive Bayes Classifier needs to multiply the posterior probabilities by the prior probabilities for each class, so the time complexity is O(C*F).

Q.3
Bayesian belief network

A- Draw the probability table for each node in the network

we use the car value table, filter rows with milage=high, and sum car values for that and finally devide that no to all numbers

mileage= hi	0.5
mileage= low	0.5

AC= working	0.5
AC= broken	0.5

engine	mealage= high	mileage= low
good	2/4	2/4
bad	2/4	2/4

Car value	engine= good, AC= working	engine= good, AC= broken	engine= bad, AC= working	engine= bad, AC= broken
hi	9+3/20= 0.6	1+5/20= 0.3	1+1/20= 0.1	0/20= 0
low	4+0/20= 0.2	2+1/20= 0.15	5+2/20= 0.35	4+2/20= 0.3

B- Use the Bayesian network to compute
P(Engine = Bad, Air Conditioner = Broken)= P(AC= broken) P(engine=bad| mileage)= 0.5*0.5=0.25

P(engine=bad| mileage)=
P(engine=bad| mileage=high) P(mileage=high) +
P(engine=bad| mileage=low) P(mileage=low)=
0.5*0.5 + 0.5*0.5 = 0.5