Homework-3

B. Deekshith
11661020
CSCE -5380

(2)

$P(Y=+\mid C_1)$	0.15	0.2	0.25	0.37	0.41	0.55	0.65	0.8	0.92	0.99
$P(Y=+\mid C_2$	0.33	0.22	0.1	0.41	0.68	0.59	0.72	0.75	0.64	0.95
Y	−	−	+	−	+	−	−	+	+	+

A) Here y is the actual class, C_1 and C_2 is predicted class

a) for $C_1 \geq 0.70$

P	N
0.8	0.15
0.92	0.2
0.99	0.25
	0.37
	0.41
	0.55
	0.65

$C_2 \geq 0.625$

P	N
0.68	0.33
0.72	0.22
0.75	0.1
0.64	0.41
0.95	0.59

~~Based~~ ~~o~~

C_1		Actual values	
		P	N
Predicted values	P	0.8 (TP) 0.92 0.99	FN
		0.25 (FP) 0.41	0.2, 0.15 0.37, 0.55 0.65

C_2		Actual values	
		P	N
Predicted values	P	0.68 0.64 0.75 0.95	
	N	0.1 0.72 (FP)	0.33 0.22 0.41 0.59

Based on Accuracy C_1 is better

Based on F both classifiers are good.

b) Accuracy $(C_1) = \frac{TP+TN}{P+N}$

$$= \frac{(0.8+0.9+0.99)+(1.92)}{5.29}$$

$$C_1 = \frac{4.63}{5.29} = 0.87$$

$$\text{Accuracy } (C_2) = \frac{TP+TN}{P+N}$$

$$= \frac{3.02+1.55}{5.39}$$

$$= 0.85$$

$$\text{f measure}(C_1) \Rightarrow \frac{2 \times \text{Precision} \times \text{Recall}}{\text{precision} + \text{Recall}}$$

$$\text{Precision} = TP/(TP+FP) = \frac{2.71}{3.37}$$

$$\text{Recall} = TP/(TP+FN) = 1$$

$$\text{F-measure } (C_1) = \frac{2 \times (0.80) \times 1}{0.80+1}$$

$$C_1 = \frac{1.6}{1.8} = 0.88$$

$$\text{Fmeasure } (C_2) = \text{precision} = 3.02/3.84$$

$$\text{Recall} = 1$$

$$c_2 = \frac{2 \times (0.786) \times 1}{1.786}$$

$$= \frac{1.57}{1.78}$$

$$c_2 = 0.880$$

3)

a) No the rules are not mutually exclusive. since the values are different from each other, we can have air conditioner and high mileage.

b) No, the rule set is not exhaustive, as there is no rule for combination of Airconditioner = Broken and mileage = medium

c) yes, in order to know wheather mileage is more important than air conditioner order is required

d) since, the rule set is not exhaustive default rule is required to cover the ~~set~~ rest of the cases

5) a)

C4.5 Rules for strength

→ easy to implement C4.5 rules to produce decision tree and also used for grouping problems

→ C4.5 can handle both categorical & continuous attributes making it unique

→ C4.5 ~~ru~~ builds models that can easily performed and also ~~concern~~ ~~seil~~ provide a clear hierarchial representation of decision rules

C4.5 The decision tree pruning in C4.5 helps avoid overfitting and improves generalization

Weakness:

-> It is not good with small training set

-> C4.5 may not find the globally optimal set of rules as it relies on a greedy, recursive approach

-> C4.5 can be sensitive to noisy data and outliers during the decision tree construct^n

Ripper:

Strengths:

It also works properly with noisy data sets because it uses a validation set to stop model overfitting

-> Ripper can handle imbalanced dataset by focusing on rule quality rather than raw accuracy.

Weakness:

Ripper may get stuck in local optima, and its rule generat^n heavily depends on the initial random select^n of rules

-> Ripper is designed for categorical data, and handling continuous attributes may require preprocessing

b) Ripper is generally better for finding high accuracy for the small classes. It focus on improving the quality of rules, which can be particularly beneficial when dealing with imbalanced dataset. It prioritizes the accurate classificat^n of the minority class, making it more suitable for scenarios where certain classes are much smaller than others.

(6.)
Given that 50 +ve & 200 -ve

R_1 : A → + (5 +ve, & 1 -ve)

R_2 = B → + (20 +ve & 5 -ve)

R_3 : C → + (50 +ve. & 40 -ve)

a) Rule accuracy
rule accuracy = p/t ratio

$R_1 = p = 5, t = 5+1 = 6$

$p/t = 5/6 = 0.833 \therefore 83\%$

$R_2 \Rightarrow p/t = 20/25 = 0.8 \therefore 80\%$

$R_3 \Rightarrow p/t = 50/90 \Rightarrow 0.555 \therefore 55.5\%$

R_1 has the highest accuracy. Therefore it is the best candidate

R_3 has the lowest accuracy therefore it is the worst candidate

b) Foil's informatⁿ gain

Foils info gain = $p_1 \times (\log_2(p_1/(p_1+n_1)) - \log_2(p_0/(p_0+n_0))$

p_0 = no. of +ve data instances before adding the candidate conditⁿ

n_0 = no. of -ve data instances before adding the candiate central

$p_0 = 50 \quad n_0 = 200$

$r_1 = p_1 = 5, \; n_1 = 1 \quad p_0 = 50, \; n_0 = 200$

$R_1 = 5[\log(5/6) - \log(50/250)] = 3.09$

$R_2 = 20[\log(20/25) - \log(50/250) = 12.04$

$R_3 = 50[\log(50/90) - \log(50/250) = 22.18$

As per this R_3 is the best condidate & R_1 is worst candidate

Scanned with CamScanner