(4.)

$$\text{Info}(D) = -\frac{6}{10}\log\frac{6}{10} - \frac{4}{10}\log\frac{4}{10}$$

$$= 0.94$$

Outlook	yes	no
Sunny	2	2
overcast	2	0
rain	2	2

$$\text{Info}_{outlook}(D) = \frac{4}{10} I(2,2) + \frac{2}{10} I(2,0) + \frac{4}{10} I(2,2).$$

$$= 0.4 + 0 + 0.4 = 0.8$$

$$\text{Info Gain}_{outlook} = 0.94 - 0.8 = 0.14.$$

Humidity	yes	no
normal	4	1.
high	2	3

$$\text{Info}_{humidity}(D) = \frac{5}{10} I(4,1) + \frac{5}{10} I(2,3)$$

$$= 0.35 + 0.27$$

$$= 0.62$$

$$\text{InfoGain}_{humity}(D) = 0.94 - 0.62$$

$$= 0.32$$

Wind	yes	no
Weak	4	1
Strong	2	3.

$$Info_{wind}(D) = \frac{5}{10} I(4,1) + \frac{5}{10} I(2,3) = \cancel{0.?} \cdot 0.62$$

3

$$InfoGain(D) = \cancel{0.94 - 0.62} = 0.94 - 0.62$$

$$= 0.32$$

∴ The root can be taken whose info gain is max.
In this case either of the two – wind or humidity can be taken as root.

(b) Possible terminating criteria of a decision tree algorithm are –

→ There are no more attributes left upon which the branching can be done further. In this case majority is taken as the leaf class.

3

→ All the tuples have the same class in a branch. In this case the branch is terminated with that class.

→ There are no tuples left.

(c) Causes of model overfitting -

When there are a lot of outliers present in the dataset and the model is trained upon those data, the model gets overfitted.

2

It is solved in decision tree by two methods -

- pre pruning - Pruning the tree before it is fully grown.
- post pruning - Pruning the decision tree branches after the tree has fully grown.

This makes the tree simplified and hence overfitting is taken care of.

(6.)

(a) Confusion matrix.

Actual \ Predicted	yes	no
yes	1	1
no	1	2

3

$$\text{precision} = \frac{TP}{TP + FP} = \frac{1}{2} = 0.5$$

$$\text{recall} = \frac{TP}{TP + FN} = \frac{1}{2} = 0.5$$

~~F-score~~

$$F_\beta = \frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$$

(c) ~~Main~~ Main purpose of ensemble method is that when a single model is used, it may have error and the same error gets repeated again and again. But when multiple models are used, all the models predict based on their training, and the decision is taken collectively. Thus the resulting prediction is much more accurate.

Adaboost algorithm :

1. Initialize the weight of all ~~to~~ the tuples to $1/d$.
2. for $i = 1$ to k do
3. Create dataset ~~d~~ D_i with replacement from D.
4. Train with D_i to create model M_i.
5. Calculate $error(M_i)$.
6. If $error(M_i) > 0.5$
7. goto step 3.
8. endif.
9. else
10. for each correctly predicted tuple, multiply ~~wt~~ weight of the tuple by $\frac{error(M_i)}{1 - error(M_i)}$.
11. normalize the weights of all the tuples.
12. end.

3

Classification

1. Initialize weight of all class to 0.
2. for $i = 1$ to k, do.
3. Predict $c = M_i(x)$.
4. Calculate $error(M_i)$.
5. Increase the weight of class c by $\frac{1 - error(M_i)}{error(M_i)}$
6. End.
7. Return the class with max weight.

(5.)

(a) $$P(\text{yes} \mid \text{overcast, high, weak}) = \frac{P(\text{overcast, high, weak} \mid \text{yes}) \times P(\text{yes})}{P(\text{overcast, high, weak})}$$

$$P(\text{no} \mid \text{overcast, high, weak}) = \frac{P(\text{overcast, high, weak} \mid \text{no}) \times P(\text{no})}{P(\text{overcast, high, weak})}$$

$$P(\text{yes}) = \frac{6}{10} \qquad P(\text{no}) = \frac{4}{10}$$

Naive Bayes:

$$P(\text{overcast, high, weak} \mid \text{yes}) = P(\text{overcast} \mid \text{yes}) \times P(\text{high} \mid \text{yes}) \times P(\text{weak} \mid \text{yes})$$

$$= \frac{1 \times 2/10}{6/10} \times \frac{2/5 \times 5/10}{6/10} \times \frac{4/5 \times 5/10}{6/10}$$

3

$$= \frac{1}{3} \times \frac{1}{3} \times \frac{2}{3} = 0.07$$

$$P(\text{overcast, high, weak} \mid \text{no}) = P(\text{overcast} \mid \text{no}) \times P(\text{high} \mid \text{no}) \times P(\text{weak} \mid \text{no})$$

$$= \frac{1/3 \times 3/11}{5/12} \times \frac{3/5 \times 5/10}{4/10} \times \frac{1/5 \times 5/10}{4/10}$$

$$= 0.04.$$

We ignore the denominator as it is const in both.

$\therefore$ $P(\text{yes} \mid \text{overcast, high, weak}) = 0.07 \times \frac{6}{10}$

$P(\text{no} \mid \text{overcast, high, weak}) = 0.04 \times \frac{4}{10}$

$\therefore$ Naive Bayes will predict "Yes".

(b) Yes, in naive Bayes, since we independently calculate all the probabilities, there is error in the prediction.

0

(c) If any feature has continuous values, then we need to ~~disce~~ discretize the attribute before applying the algorithm.

0

2. (b) Bread (1), Butter (2), Milk (3), Jelly (4), Coke (5).

C_0 : $\{1\}$, $\{2\}$, $\{3\}$ $\{4\}$ $\{5\}$

$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{5\}$
5/6	3/6	3/6	1/6	2/6

F_0 : $\{1\}$ $\{2\}$ $\{3\}$ $\{5\}$

C_1 : $\{1,2\}$, $\{1,3\}$, $\{1,5\}$, $\{2,3\}$, $\{2,5\}$, $\{3,5\}$

$\{1,2\}$	$\{1,3\}$	$\{1,5\}$	$\{2,3\}$	$\{2,5\}$	$\{3,5\}$
3/6	2/6	1/6	1/6	0	1/6

2

F_1 : $\{1,2\}$, $\{1,3\}$.

C_2 : ϕ

~~C_2 : 1, 2, 3~~

[since $\{1,2\}$ and $\{1,3\}$ are present, but $\{2,3\}$ is not present, $\{1,2,3\}$ cannot be a candidate].

(c) Drawbacks of apriori algorithms :-

- It does not takes into account the ~~number~~ count of ~~item is~~ particular item is present in a transaction.

- It is a very time complex algorithm.

0

(a) For an association rule,

$$A \longrightarrow B$$

$$\text{support} = \frac{(A \vee B).\text{count}}{n}$$

n = total no. of transactions.

1

$$\text{confidence} = \frac{B.\text{count}}{A.\text{count}}$$

An itemset is called frequent itemset when it's support is greater than min. support.