Assignment

Applied Generative AI and Machine Learning

Name : Adithya Rao Kalathur

Reg No. : 251100610012

Branch : Computer Science and Engineering (M.E. CSE)

Email : adithya.mssmpl2025@learner.manipal.edu

Signature : Adithya Rao Kalathur

Question 1 -

Obtain DT for the following data

Instance	Attribute 1	Attribute 2	Class
1	T	T	+
2	T	T	+
3	T	F	−
4	F	F	+
5	F	T	−
6	F	T	−

Answer -

Class Distribution:

Total instances = 6

Positive (+) = 3

Negative (−) = 3

$$\text{Entropy}(S) = p_{+ve} \log_2(p_{+ve}) - p_{-ve} \log_2(p_{-ve})$$

$$p_{+ve} = \frac{3}{6} = 0.5$$

$$p_{-ve} = \frac{3}{6} = 0.5$$

$$\text{Entropy}(S) = -0.5\log_2(0.5) - 0.5\log_2(0.5) = 1$$

$$\text{Entropy}(S) = 1$$

Calculating Information Gain for each attribute

a) Attribute_1 -

T - 3 instances = 2+ve, 1-ve

$$\text{Entropy} = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

F - 3 instances = 1+ve, 2-ve

$$\text{Entropy} = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.918$$

So weighted average entropy

$$\text{Entropy}(\text{Attribute\_1}) = \frac{3}{6}(0.918) + \frac{3}{6}(0.918) = 0.918$$

$$\text{Information gain}(\text{Attribute\_1}) = 1 - 0.918 = 0.082$$

b) Attribute_2 -

T - 4 instances = 2+ve, 2-ve

$$\text{Entropy} = \frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

F - 2 instances = 1+ve, 1-ve

$$\text{Entropy} = \frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

So weighted average entropy

$$\text{Entropy}(\text{Attribute\_2}) = \frac{4}{6}(1) + \frac{2}{6}(1) = 1$$

Information Gain (Attribute_2) = 1 - 1 = 0

Choosing best Attribute

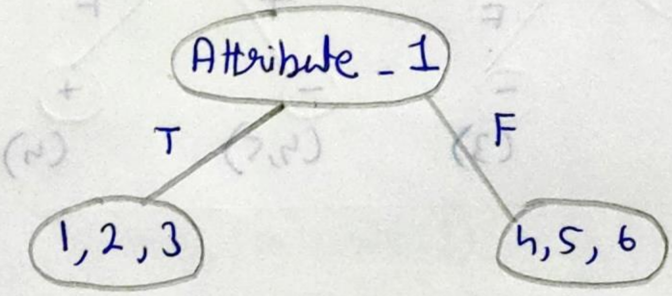
IG(Attribute_1) = 0.082
IG (Attribute_2) = 0

IG (Attribute_1) > IG (Attribute_2)

Therefore Attribute_1 is the root node.

Building subtrees

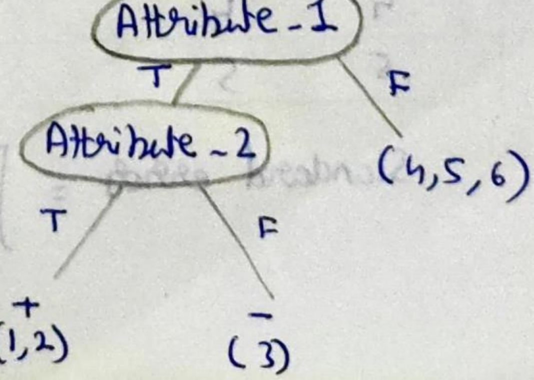
current DT -



New table for T of attribute_1

Instance	Attribute 2	class
1	T	+
2	T	+
3	F	-

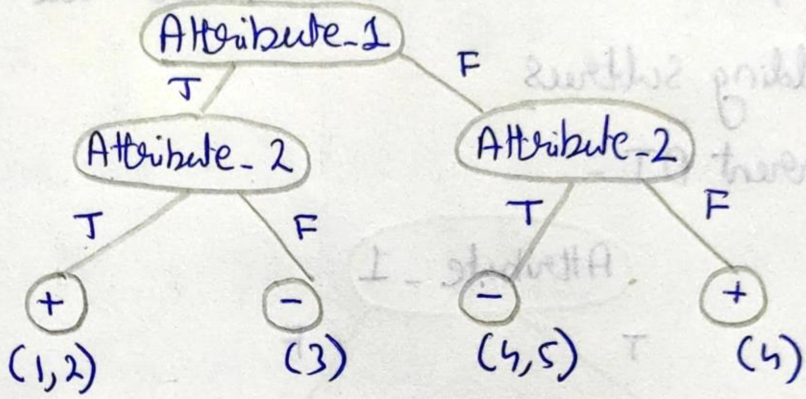
The new tree from the table becomes



New table for F of Attribute-1

Instance	Attribute_2	class
4	F	+
5	T	−
6	T	−

Final Decision Tree



Question 2 -

Obtain a line that best fit the sample data given in the table. Evaluate the model by finding the standard error

(x) IDV	Marks scored (y) DV			
1	2			
2	4 ~~3~~			
3	5			
4	4			
5	5			

Standard error $= \sqrt{\dfrac{\Sigma(\hat{y}-y)^2}{n-2}}$

Answer -

x	y	x^2	xy	$\hat{y}$
1	2	1	2	2.8
2	4	4	8	3.4
3	5	9	15	4
4	4	16	16	4.6
5	5	25	25	5.2
sum=15	20	55	66	
Avg($\bar{x}$) = 3	$\bar{y}=4$	$\overline{x^2}=11$	$\overline{xy}=13.2$	

Equation of the line,

$$\hat{y} = a_0 + a_1 x + e, \text{ where e is error}$$

$$a_0 = \bar{y} - a_1(\bar{x})$$

$$a_1 = \frac{(\overline{xy}) - \bar{x}(\bar{y})}{(\overline{x^2}) - (\bar{x})^2}$$

From table,

$$a_1 = \frac{13.2 - 3(4)}{11 - 3^2} = 0.6 \quad -①$$

from table & ①

$$a_0 = 4 - 0.6(3) = 2.2 \quad - ②$$

from ① & ②

$$\hat{y} = 2.2 + 0.6(x_n) + e$$

↳ equation of the line

$$\text{Standard error} = \sqrt{\frac{\Sigma(\hat{y}_i - y)^2}{n-2}}$$

Solving inner terms

$$\Sigma(\hat{y_i} - y)^2 = (2.8-2)^2 + (3.4-4)^2 + (4-5)^2 + (4.6-4)^2 + (5.2-5)^2$$

$$= 2.4 \text{ --- } (3)$$

From (3)

$$\text{Standard Error} = \sqrt{\frac{2.4}{5.2}} = 0.8944$$

Question - 3 -

- Explain with example about batch, minibatch & stochastic gradient descent.
- Compare batch, minibatch & stochastic gradient descent
- Different evaluation metrics for regression.

Answer -

1) **Batch Gradient Descent** - uses the entire training dataset to compute gradient for each parameter update.

Formula - $\theta = \theta - \alpha \nabla J(\theta)$

Algorithm -

For each epoch:

-> calculate cost $J(\theta)$ using all training samples

-> compute gradient : $\nabla J(\theta) = (1/m) \Sigma(h(x) - y)$

-> update parameter : $\theta = \theta - \alpha \nabla J(\theta)$

Example - Dataset has 1000 samples

- calculates gradient using all 1000 samples
- Makes 1 parameter update per epoch
- Smooth convergence but slow

Mini-Batch Gradient Descent - uses small batches of training data to compute gradient.

Formula: $\theta = \theta - \alpha \nabla J(\theta\_batch)$

Algorithm:

→ For each epoch:

→ Divide data into batches of size B

→ For each batch:

1) calculate cost using B samples
2) compute gradient for each batch
3) update parameters

Example - Dataset has 1000 samples with batch size = 100

- Creates 10 batches of 100 samples each
- Makes 10 parameter updates per epoch
- Good balance of speed and stability

Stochastic Gradient Descent (SGD) - uses one training sample at a time to compute gradient.

Formula - $\theta = \theta - \alpha \nabla J(\theta_i)$ where i is single sample

Algorithm:

For each epoch:

For each sample i:

1) calculate cost using sample i only
2) compute gradient for single sample
3) update parameters immediately

Example - Dataset has 1000 samples

- uses 1 sample at a time

- Makes 1000 parameter updates per epoch
- Fast but noisy convergence

2) Comparision Table:

Aspect	Batch GD	Mini-Batch GD	Stochastic GD
Data per update	All samples (m)	Batch size (k)	1 sample
updates per epoch	1	m/k	m
convergence	smooth	moderately smooth	Noisy
Speed	Slow	Medium	Fast
Memory usage	High	Medium	Low
Accuracy	High	High	medium
Best for	Small dataset	Large dataset	online learning

Example - Training neural network on 10,000 images
- Batch GD: 1 update / epoch, uses all 10000 images
- Mini-Batch GD: 100 updates / epoch (batch size = 100)
- SGD: 10000 updates / epoch, 1 image at a time

3) Regression Evaluation Metrics

-) Mean Absolute Error (MAE) - Easy to interpret & less sensitive to outliers

Formula - $MAE = (1/n) \sum_{i=1}^{n} |y_i - \hat{y}_i|$

Examples - Actual : [10, 20, 30, 40]
Predicted : [12, 18, 28, 45]
Errors : [2, 2, 2, 5]
$MAE = (2+2+2+5)/4 = 2.75$

→ Mean Squared Error (MSE) – Penalizes large errors more & is sensitive to outliners

Formula – $MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

Example – Actual : [10, 20, 30, 40]
Predicted : [12, 18, 28, 45]
Errors : [2, 2, 2, 5]
Squared Errors : [4, 4, 4, 25]
MSE = (4 + 4 + 4 + 25) / 4 = 9.25

→ Root Mean Squared Error (RMSE) – Same unit as target variable & easier to interpret than MSE

Formula – $RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$

Example – $RMSE = \sqrt{9.25} = 3.04$

→ Mean Absolute Percentage Error (MAPE) –

Formula – $MAPE = (100/n) \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$

Example – Percentage errors : [20%, 10%, 6.7%, 12.5%]
MAPE = (20 + 10 + 6.7 + 12.5) / 4 = 12.3%

~~Rodeck~~

→ R – Squared (coefficient of Determination) – Measures how much variance in data is explained by model.

- $R^2 = 1$ → perfect fit, $R^2 = 0$ → no improvement over mean.

Formula – $R^2 = 1 - (SSres / SStot)$

where :

- $SSres = \Sigma (y_i - \hat{y}_i)^2$ (sum of squared residuals)

• $SS_{tot} = \Sigma (y_i - \bar{y})^2$ (Total sum of squares)

Example - Actual mean: $\bar{y} = 25$

$SS_{tot} = (10-25)^2 + (20-25)^2 + (30-25)^2 + (40-25)^2$

$= 225 + 25 + 25 + 225$

$= 500$

$SS_{res} = 4 + 4 + 4 + 25 = 37$

$R^2 = 1 - (37/500) = 1 - 0.074 = 0.926$

→ **Adjusted R-Squared** -

Formula: $R^2_{adj} = 1 - [(1-R^2)(n-1)/(n-p-1)]$

where: n = samples, p = predictors

Example - $n = 4$, $p = 1$

$R^2_{adj} = 1 - [(1-0.926)(4-1)/(4-1-1)]$

$= 1 - [0.074 \times 3/2]$

$= 0.889$

Question - 4 -

Table shows the output y, and the probability of the predicted output y-pred.

Find the predicted output class for threshold 0.6

calculate the following

→ True positive

→ True negative

→ False positive

→ False negative

→ Accuracy, precision, recall, and f-1 score.

Answers -

Given Table

Actual (y)	Predicted Probability (y-pred)
0	0.5
1	0.9
0	0.7
1	0.7
1	0.3
0	0.4
1	0.5

Applying the threshold (0.6)

if y-pred >= 0.6 → predicted class = 1,

else → predicted class = 0

y	y-pred	Predicted class	Type
0	0.5	0	True Negative (TN)
1	0.9	1	True Positive (TP)
0	0.7	1	False Positive (FP)
1	0.7	1	True Positive (TP)
1	0.3	0	False Negative (FN)
0	0.4	0	True Negative (TN)
1	0.5	0	False Negative (FN)

TP = 2, TN = 2, FP = 1, FN = 2

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2+2}{2+2+1+2} = \frac{4}{7} \approx 0.571$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{2}{2+1} = \frac{2}{3} \approx 0.667$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{2}{2+2} = \frac{2}{4} = 0.5$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times 0.667 \times 0.5}{0.667+0.5} = \frac{0.667}{1.167}$$

$$\approx 0.571$$

Question -5 -

Consider a regression problem where you are trying to predict the number of readers of the article, & let's say your test set has two observations $n = 2$.

If actual number of readers, $y_j = [10, 5]$ & your model predicts, $\hat{y}_j = [8, 6]$ readers then find the following

$$R\,\text{square} = 1 - \frac{(y_i - \hat{y}_i)^2}{(y_i - y_{mean})^2}$$

1. MAE
2. RMSE
3. R Squared / coefficient of determination

Answers

Given Data

Observation	Actual (y)	Predicted ($\hat{y}$)
1	10	8
2	5	6

$y = [10, 5]$, $\hat{y} = [8, 6]$

No. of samples $n = 2$

1. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i|$$

$$= \frac{|10-8| + |5-6|}{2} = \frac{2+1}{2} = 1.5$$

2. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$$

$$= \sqrt{\frac{(10-8)^2 + (5-6)^2}{2}} = \sqrt{\frac{4+1}{2}} = \sqrt{2.5} = 1.5811$$

3. R^2 (coefficient of Determination)

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

$$\bar{y} = \frac{10+5}{2} = 7.5$$

$$\sum (y_i - \hat{y}_i)^2 = (10-8)^2 + (5-6)^2 = 4+1 = 5$$

$$\sum (y_i - \bar{y})^2 = (10-7.5)^2 + (5-7.5)^2 = 2.5^2 + (-2.5)^2 = 6.25+6.25$$
$$= 12.5$$

$$\therefore R^2 = 1 - \frac{5}{12.5} = 1-0.4 = 0.6$$

Question -6-

Use K-means clustering algorithm to divide the following data points into 2 clusters.

x_1	1	2	2	3	4	5
x_2	1	1	3	2	3	5

Answers -

Lets choose the 1^{st} & last points as initial centroids.

$C1 = (1,1)$
$C2 = (5,5)$

Computing distances to initial centroids (iteration 1) -

Point	d to $C_1 = \sqrt{(x-1)^2+(y-1)^2}$	d to $C_2 = \sqrt{(x-5)^2+(y-5)^2}$	Nearest
P1 (1,1)	$\sqrt{(1-1)^2+(1-1)^2} = 0.0$	$\sqrt{(1-5)^2+(1-5)^2} = 5.657$	C1
P2 (2,1)	$\sqrt{(2-1)^2+(1-1)^2} = 1.0$	$\sqrt{(2-5)^2+(1-5)^2} = 5.00$	C1
P3 (2,3)	$\sqrt{(2-1)^2+(3-1)^2} = 2.236$	$\sqrt{(2-5)^2+(3-5)^2} = 3.606$	C1
P4 (3,2)	$\sqrt{(3-1)^2+(2-1)^2} = 2.236$	$\sqrt{(3-5)^2+(2-5)^2} = 3.606$	C1
P5 (4,3)	$\sqrt{(4-1)^2+(3-1)^2} = 3.606$	$\sqrt{(4-5)^2+(3-5)^2} = 2.236$	C2
P6 (5,5)	$\sqrt{(5-1)^2+(5-1)^2} = 5.657$	$\sqrt{(5-5)^2+(5-5)^2} = 0.0$	C2

Cluster assignment (iteration 1) -

C1 = (1,1) (2,1) (2,3) (3,2)
C2 = (4,3) (5,5)

Computing New Centroids.

Cluster	sum x1	sum x2	n	centroid = (mean x1, mean x2)
C1	1+2+2+3 = 8	1+1+3+2 = 7	4	(8/4, 7/4) = (2, 1.75)
C2	4+5 = 9	3+5 = 8	2	(9/2, 8/2) = (4.5, 4)

Distances to updated centroids (iteration 2)

Point	dis C1 = $\sqrt{(x-2)^2+(y-1.75)^2}$	dis C2 = $\sqrt{(x-4.5)^2+(y-4)^2}$	Nearest
P1(1,1)	$\sqrt{(1-2)^2+(1-1.75)^2} = 1.250$	$\sqrt{(1-4.5)^2+(1-4)^2} = 4.610$	C1
P2(2,1)	$\sqrt{(2-2)^2+(1-1.175)^2} = 0.750$	$\sqrt{(2-4.5)^2+(1-4)^2} = 3.905$	C1
P3(2,3)	$\sqrt{(2-2)^2+(3-1.175)^2} = 1.250$	$\sqrt{(2-4.5)^2+(3-4)^2} = 2.693$	C1
P4(3,2)	$\sqrt{(3-2)^2+(2-1.175)^2} = 1.031$	$\sqrt{(3-4.5)^2+(2-4)^2} = 2.500$	C1
P5(4,3)	$\sqrt{(4-2)^2+(3-1.175)^2} = 2.359$	$\sqrt{(4-4.5)^2+(3-4)^2} = 1.118$	C2
P6(5,5)	$\sqrt{(5-2)^2+(5-1.175)^2} = 4.423$	$\sqrt{(5-4.5)^2+(5-4)^2} = 1.118$	C2

cluster assignment (Iteration 2) -

C1 = (1,1), (2,1), (2,3), (3,2)

C2 = (4,3), (5,5)

No change in assignment → algorithm has converged

Final results

cluster	Points in cluster	Final centroid (x1,x2)
C1	(1,1)(2,1) (2,3) (3,2)	(2.0, 1.75)
C2	(4,3) (5,5)	(4.5, 4.0)