5)(a)

- Selecting the ~~root~~ attribute for split

Outlook (14)

	Sunny	Overcast	Rain
Yes	2	4	3
No	3	0	2
	0.97	0	0.97

$\Rightarrow \frac{5}{14}(0.97) + 0 + \frac{5}{14}(0.97)$

$= 0.693$ ✓

Climate (14)

	Hot	Mild	Cool
Yes	2	4	3
No	2	2	1
	1	0.92	0.81

$\Rightarrow \frac{4}{14}(1) + \frac{6}{14}(0.92) + \frac{4}{14}(0.81)$

$= 0.911$

Humidity

	Normal	High
Yes	6	3
No	1	4
	0.59	0.98

$\Rightarrow \frac{7}{14}(0.59) + \frac{7}{14}(0.98)$

$= 0.785$

Wind

	Weak	Strong
Yes	6	3
No	2	3
	0.81	1

$\Rightarrow \frac{8}{14}(0.81) + \frac{6}{14}(1)$

$= 0.891$

- OUTLOOK GIVES BEST SPLIT

DT ⇒ Outlook (14)

Sunny → N1 (5)

Overcast → Yes (4) (no more splitting)

Rain → N2 (5)

Node N1 (Outlook = Sunny)

Climate (5)

	Hot	Mild	Cool
Yes	0	1	1
No	2	1	0
	0	1	0

$\Rightarrow \frac{2}{5}(1) + 0 + 0 = 0.4$

Humidity (5)

	Normal	High
Yes	2	0
No	0	3
	0	0

$\Rightarrow 0$ ✓

Wind (5)

	Weak	Strong
Yes	1	1
No	2	1
	0.92	1

$\Rightarrow \frac{3}{5}(0.92) + \frac{2}{5}(1) = 0.952$

DT ⇒ N1 (Sunny)

Humidity = Normal → Yes (2)

Humidity = High → No (3)

N2 (Outlook = Rain)

Climate (5)

	Hot	Mild	Cool
Yes	0	2	1
No	0	1	1
	0	0.92	1

$\Rightarrow 0 + \frac{3}{5}(0.92) + \frac{2}{5}(1)$

$= 0.952$

Humidity

	Normal	High
Yes	2	1
No	1	1
	0.92	1

$\Rightarrow \frac{3}{5}(0.92) + \frac{2}{5}(1)$

$= 0.952$

Wind

	Weak	Strong
Yes	3	0
No	0	2
	0	0

$\Rightarrow 0$ ✓

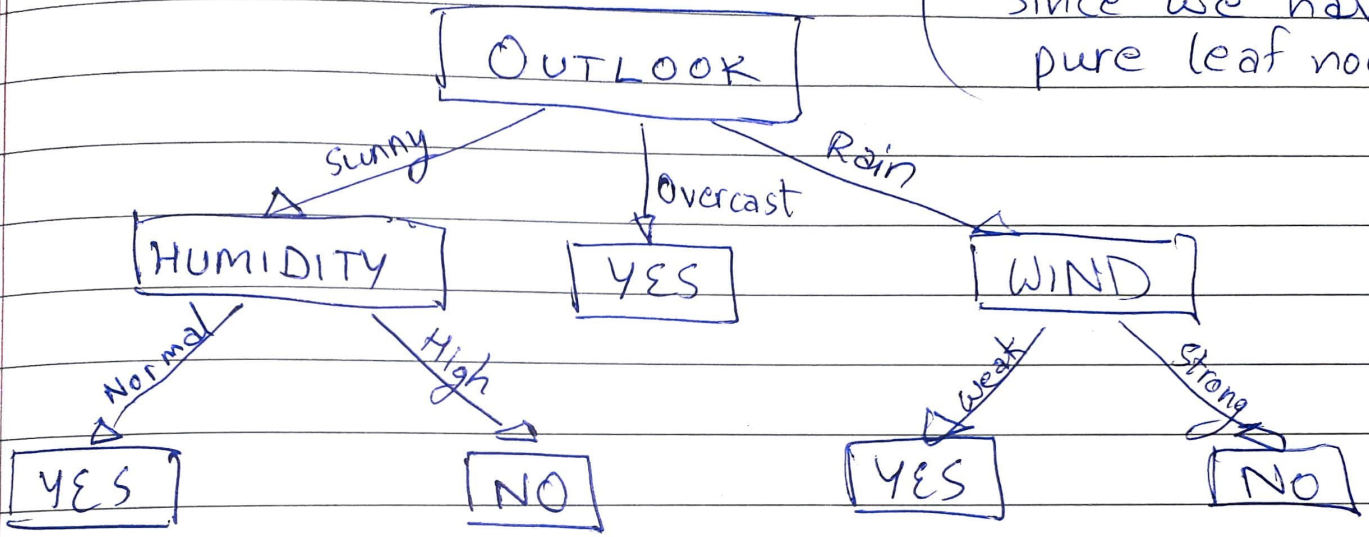
DT

N2 (Outlook = Rain)

- Wind = Weak → Yes (3)
- Wind = Strong → No (2)

⇒ FINAL DT will be :-

(* Accuracy = 100% Since we have pure leaf nodes)



5b) Outlook = Sunny, Climate = Cool, Humidity = High, Wind = ~~Weak~~ Strong, Play = ~~No~~ Yes, Day = D15.

Adding the above row, ~~we~~ the distribution will change as follows :-

Outlook
↓ Sunny
~~Play~~ N1

Also Humidity will be selected to split N1. not

Humidity

	Normal	High
Yes	2	1
No	0	3
	0	0.81

$\Rightarrow \frac{4}{6}(.81)$

$= 0.54$

Wind

	Weak	Strong
Yes	1	2
No	2	1
	0.92	0.92

$\Rightarrow \frac{3}{6}(.92) \times 2$

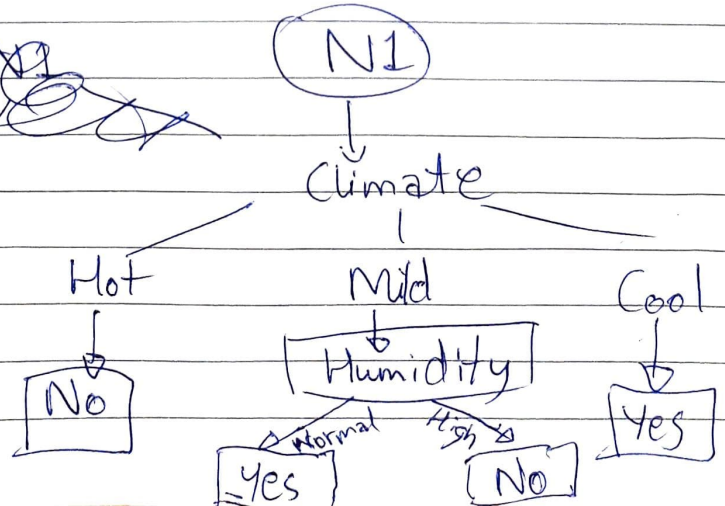
$= 0.92$

Climate

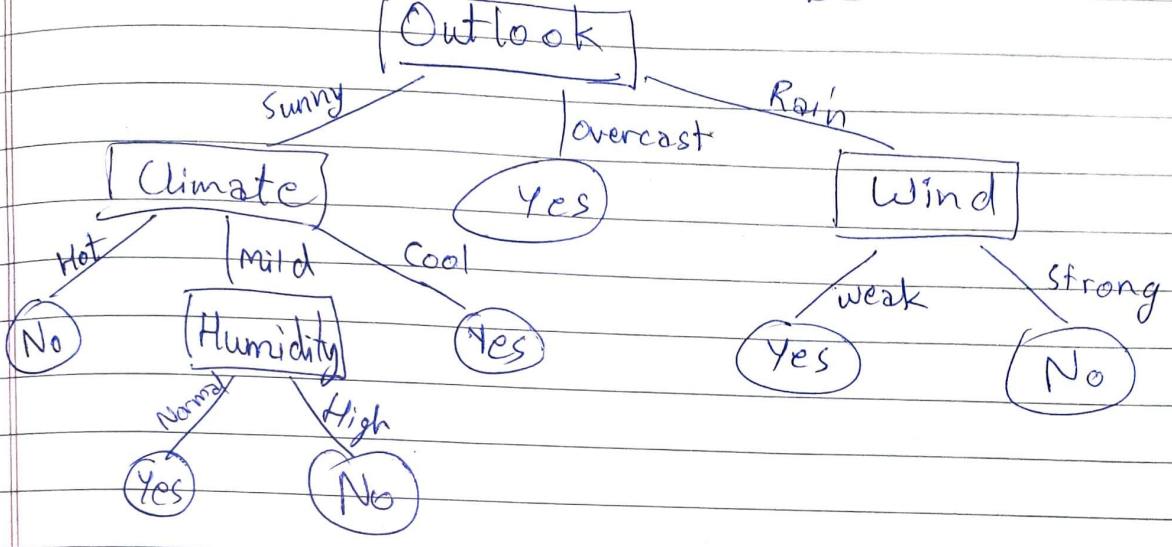
	Hot	Mild	Cool
Yes	0	1	2
No	2	1	0
	0	1	0

$\Rightarrow \frac{2}{6}(1) = 0.33$ ✓

∴ ~~N1~~



The new DT would have been:-



(5c) D1 - D7 (Training)

Outlook (7)

	Sunny	Overcast	Rain
Yes	0	2	2
No	2	0	1
	0	0	0.92

$\Rightarrow (0.92)\frac{3}{7} = \boxed{0.394}$ ✓

Humidity (7)

	Normal	High
Yes	2	2
No	1	2
	0.92	1

$\Rightarrow 0.92\left(\frac{3}{7}\right) + 1\left(\frac{4}{7}\right) = \boxed{0.966}$

Wind (7)

	Weak	Strong
Yes	3	1
No	1	2
	0.81	0.92

$\Rightarrow \frac{4}{7}(0.81) + \frac{3}{7}(0.92)$

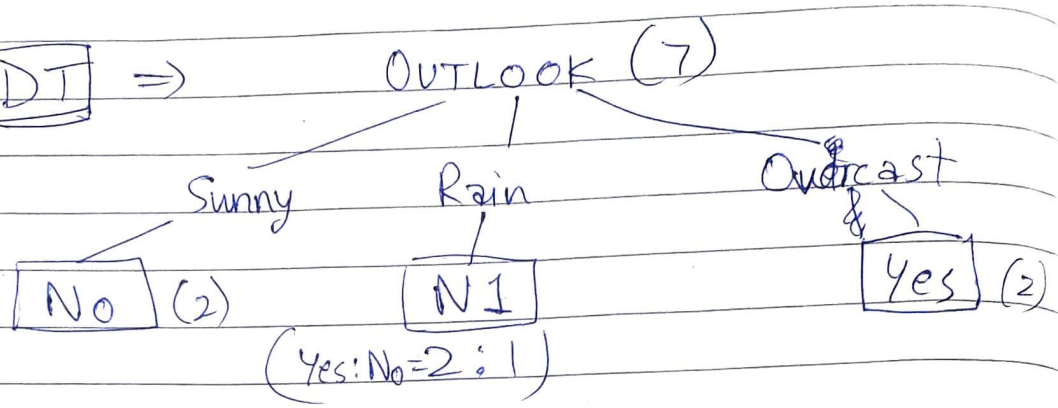
$= \boxed{0.857}$

Climate

	Hot	Mild	Cool
Yes	1	1	2
No	2	0	1
	0.92	0	0.92

$\Rightarrow \frac{6}{7}(0.92) = \boxed{0.788}$

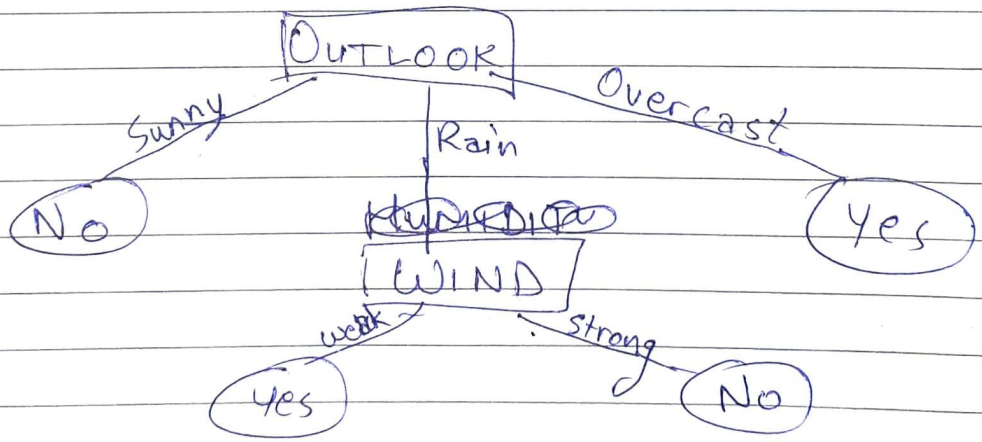
∴ DT ⇒



Outlook = Rain : N1 splitting

By observation, we see that
~~where~~ ∀ records with outlook = Rain,
Wind = Weak ⇒ Play = Yes
and Wind = Strong ⇒ Play = No

∴ DT becomes



* Since leaf nodes are pure, train accuracy = 100%

* Evaluating test accuracy

Node	D8	D9	D10	D11	D12	D13	D14
yPred	No	No	Yes	No	Yes	Yes	No
yTrue	No	Yes	Yes	Yes	Yes	Yes	No
	✓	✗	✓	✗	✓	✓	✓

* Test accuracy = $\frac{5}{7} \times 100 = \boxed{71.43\%}$

* We did not have enough training data, plus we let the tree build fully, thus overfitting on the training data. picking up any noisy sample available also, leading to a bad test accuracy in comparison to the train accuracy.
* The small amount of data did not allow the model to understand the true relationship.

5(d) There are two possible pruning strategies.

(i) Pre-pruning :- Stop the algo before fully grown tree, typically if all instances belong to the same class or if all attribute values are the same or by some user specified threshold parameters.

(ii) Post-pruning :- Generally we do not know when to stop the algorithm. Therefore, first we let the tree grow to its entirety, and the apply pruning.

Since pruning is to avoid overfitting, we should either work with a validation set error or some generalized error measurements.

The following two techniques can be used :-

→ Subtree replacement : Trim the nodes in a bottom-up manner, and if the performance improves, replace the subtree with a leaf node, having class label of the majority class

→ Subtree raising : Replace the subtree with the branch which is most frequently taken if this improves the ~~generalized~~ performance.

*These strategies would work since the performance measure is taking into account, both seen and unseen data performance.

(6) First Order Markov model says that Probability of an n^{th} event depends only on the $(n-1)^{th}$ event and the events before that have zero weightage in the probability.

We are supposed to find $P(w_3 | w_1, w_2, w_4)$ which reduces to $P(w_3 | w_2, w_4)$ by the 1st-order Markov assumption.

The order of these events is as follows:-

$$w_2 \rightarrow w_3 \rightarrow w_4$$

∴ Bayes rule gives

$$P(w_3 | w_2, w_4) = \frac{P(w_4 | w_2, w_3) \cdot P(w_3 | w_2)}{\sum_{\forall \text{ possible } w_3} P(w_4 | w_2, w_3) \cdot P(w_3 | w_2)}$$

By Markov assumption,

$$P(w_4 | w_2, w_3) = P(w_4 | w_3)$$

$$\therefore P(w_3 = \text{tough} | w_2, w_4) \quad (w_2 = \text{course}, w_4 = \text{course})$$

$$= \frac{P(w_4 | w_3 = \text{tough}) \cdot P(w_3 = \text{tough} | w_2)}{P(w_4 | w_3 = \text{tough}) \cdot P(w_3 = \text{tough} | w_2) + P(w_4 | w_3 = \text{course}) \cdot P(w_3 = \text{course} | w_2)}$$

$$= \frac{(0.3)(0.5)}{(0.3)(0.5) + (0.5)(0.5)} = \frac{3}{8} = 0.375$$

$$\boxed{P(w_3 = \text{tough} | w_2, w_4) = 0.375}$$

$$\boxed{P(w_3 = \text{course} | w_2, w_4) = 1 - 0.375 = 0.625}$$

(7) DT : Decision Tree ; LR : Logistic Regression

(a) → Unlike LR, DT does not assume all the features to be independent.

→ But, the biggest advantage is that once built, DT is easy to understand and gives direct pattern for data analysis. It can be easily converted to an SQL syntax to apply on big datasets.

→ DT works better when there are more than a single decision boundary, but parallel to the feature axes.

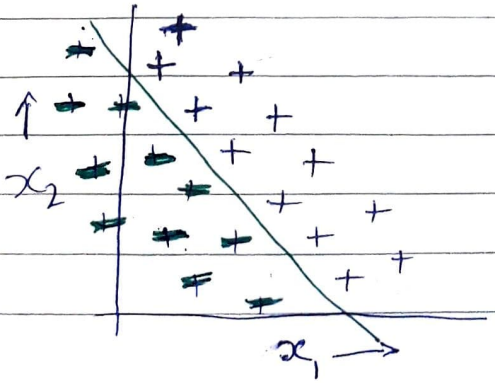
7(b) → LR, unlike DT, can draw any decision boundary, which need not be parallel to the feature axes.

→ DT has relatively high cost in terms of memory and time. Splitting a node involves lots of comparisons, that is kind of a bruteforce.

→ DT when scaled up to very high depths are very much prone to overfitting.

7(c) $$y = \begin{cases} +1 & w_1x_1 + w_2x_2 + b \geq 0 \\ -1 & w_1x_1 + w_2x_2 + b \leq 0 \end{cases}$$

7(e) n :- no. of data samples



In the worst case, each data sample gets classified with the help of 2 unique decision boundaries.

In the worst case, thus, we will have n+1 boundaries on each of the axes.

~~Thus, there will be $(n+1)^2$ leaf nodes.~~

* Imagine first of all x_1 only is used for creating boundaries leading to (n+1) leaves.
* Each of these (n+1) leaf nodes would further split to create (n+1) × (n+1) leaf nodes in total.

Thus, a bound on depth can be found.
Let ~~depth~~ Level 0 be the root.
Let level i have 2^i nodes considering only binary splits.

$$\therefore 2^i = (n+1)^2$$

$$\Rightarrow i = \boxed{2\log_2(n+1)} \leq 2\log(2n)$$

$$(\text{Depth}) = \boxed{O(\log_2(n))}$$

7(d) ~~Draw~~ (continuing from (e) part, if x_2 were linearly dependent on x_1, then the boundary for x_2 could be derived from x_1.

* After (n+1) boundaries by x_1, we would only need 1 additional node per leaf to create the x_2 boundary.
* Thus 2(n+1) leaf nodes

Level i: $2^i = 2(n+1)$

$$\Rightarrow i = \boxed{\log_2(n+1)+1} \leq 2\log_2(2n)$$

$$\therefore \boxed{\text{Depth} = O(\log(n))}$$

(8) $X = \langle X_1, X_2, X_3, \ldots, X_n \rangle$ X_i = Boolean Var.
$P(Y=1) = \pi$

To P.: $P(Y=1|X) = \dfrac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{n} w_i X_i\right)}$

Pf: Let $\theta_{i1} = P(X_i = 1 | Y = 1)$ $(\forall\, 1 \le i \le n)$
$\Rightarrow P(X_i = 0 | Y = 1) = 1 - \theta_{i1}$ $(\forall\, 1 \le i \le n)$

Similarly, $P(X_i = 1 | Y = 0) = \theta_{i0}$ (")
and, $P(X_i = 0 | Y = 0) = 1 - \theta_{i0}$ (")

From the naive bayesian model, we have,

$$P(Y=1|X) = \frac{P(X|Y=1) \cdot P(Y=1)}{\sum_{c} P(X|Y=c) \cdot P(Y=c)}$$

$$= \frac{1}{\frac{P(X|Y=0) \cdot P(Y=0)}{P(X|Y=1) \cdot P(Y=1)} + \frac{P(X|Y=1) \cdot P(Y=1)}{P(X|Y=1) \cdot P(Y=1)}}$$

$$= \frac{1}{1 + \frac{1-\pi}{\pi} \cdot \frac{P(X|Y=0)}{P(X|Y=1)}}$$

* $P(X_i | Y = c)$

$X_i = 0$: $(1 - \theta_{ic})$ $\quad$ $X_i = 1$: (θ_{ic})

$$(1 - \theta_{ic})^{1 - X_i} (\theta_{ic})^{X_i}$$

$$= \frac{1}{1 + \exp\left(\ln\left(\frac{1-\pi}{\pi} \cdot \frac{P(X|Y=0)}{P(X|Y=1)}\right)\right)}$$

$$= \frac{1}{1 + \exp(z)} \quad \text{(say)}$$

$$z = \ln\left(\frac{1-\pi}{\pi}\right) + \ln\left(\frac{P(X|Y=0)}{P(X|Y=1)}\right)$$

$$= \ln\left(\frac{1-\pi}{\pi}\right) + \ln\left(\frac{\prod_{i=1}^{n} P(X_i|Y=0)}{\prod_{i=1}^{n} P(X_i|Y=1)}\right)$$

$$= \ln\left(\frac{1-\pi}{\pi}\right) + \sum_{i=1}^{n} \ln\left[\frac{(1-\theta_{i0})^{1-x_i}\,\theta_{i0}^{x_i}}{(1-\theta_{i1})^{1-x_i}\,\theta_{i1}^{x_i}}\right]$$

$$= \ln\left(\frac{1-\pi}{\pi}\right) + \sum_{i=1}^{n} \ln\left(\frac{1-\theta_{i0}}{1-\theta_{i1}}\right) + \sum_{i=1}^{n} \ln\left[\left(\frac{1-\theta_{i1}}{1-\theta_{i0}}\right)\left(\frac{\theta_{i0}}{\theta_{i1}}\right)\right]^{x_i}$$

$$= \underbrace{\ln\left(\frac{1-\pi}{\pi}\right) + \sum_{i=1}^{n} \ln\left(\frac{1-\theta_{i0}}{1-\theta_{i1}}\right)}_{\omega_0} + \sum_{i=1}^{n} x_i \underbrace{\ln\left[\left(\frac{1-\theta_{i1}}{\theta_{i1}}\right)\left(\frac{\theta_{i0}}{1-\theta_{i0}}\right)\right]}_{\omega_i}$$

$$\Rightarrow \omega_0 = \ln\left(\frac{1-\pi}{\pi}\right) + \sum_{i=1}^{n} \ln\left(\frac{1-\theta_{i0}}{1-\theta_{i1}}\right)$$

$$\Rightarrow \omega_i = \ln\left[\left(\frac{1-\theta_{i1}}{\theta_{i1}}\right)\left(\frac{\theta_{i0}}{1-\theta_{i0}}\right)\right]$$

$$= \ln\left[\frac{P(X_i=0|Y=1)}{P(X_i=1|Y=1)} \cdot \frac{P(X_i=1|Y=0)}{P(X_i=0|Y=0)}\right] \quad (\forall\, 1 \le i \le n)$$

$$\Rightarrow P(Y=1|X) = \frac{1}{1+\exp\left(\omega_0 + \sum_{i=1}^{n} \omega_i X_i\right)}$$

Hence Proved