UNIT–2 : Data Preprocessing & Concept Learning (PYQ Solution)

PART–A Answers (≈70 Words Each)

Q1. What are missing values? Name two ways to handle them. (Expected)

Missing values are entries that were not recorded or are unavailable in a dataset; in R they appear as NA. They arise from skipped survey questions, sensor failure or data entry errors.

Two ways to handle them:

1. **Deletion: remove rows with missing values using na.omit(df) when only a few are missing.**
2. **Imputation: fill with the mean, median or mode, for example df[["age"]] [is.na(df[["age"]])] <- median(df[["age"]], na.rm = TRUE).**

Q2. Define VC dimension. (Expected)

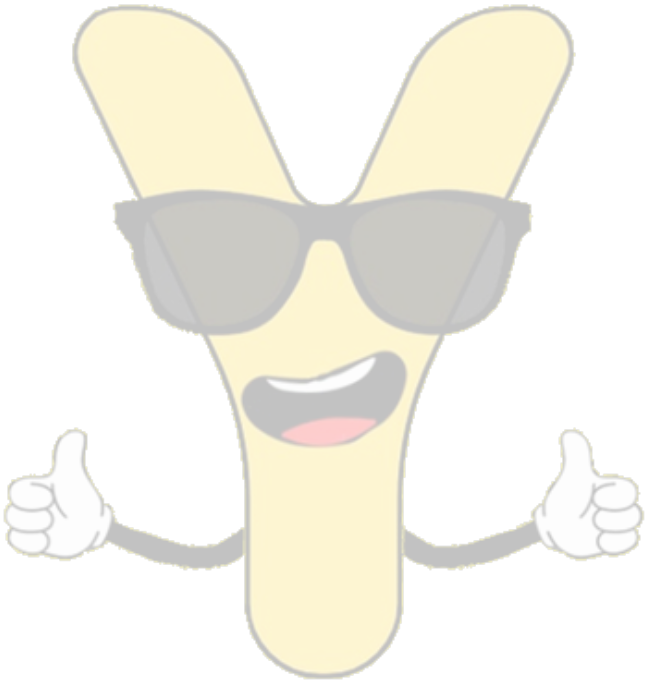
The VC (Vapnik-Chervonenkis) dimension of a hypothesis class is the size of the largest set of points that the class can shatter, meaning that for every one of the 2^n possible labellings some hypothesis in the class separates the points correctly.

Example: a straight line in 2D can shatter 3 points but not 4 (XOR pattern), so its VC dimension is 3. A higher VC dimension means a more flexible model with more risk of overfitting.

Q3. What is a hypothesis space and a version space? (Expected)

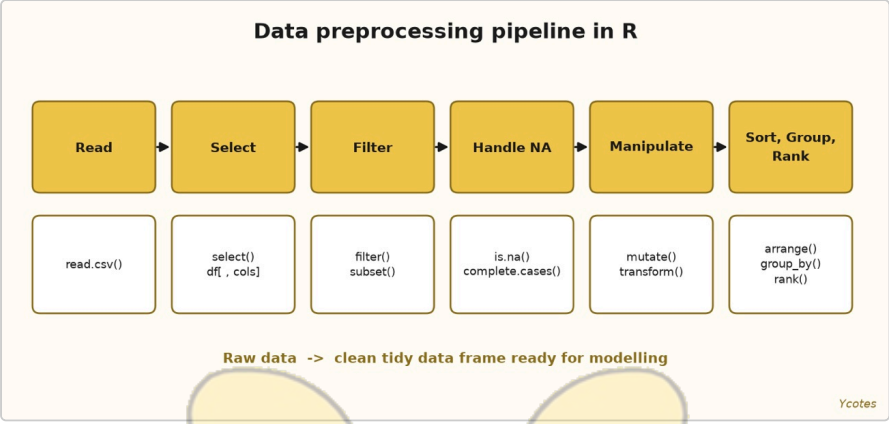
Hypothesis space (H): the set of all hypotheses (candidate rules) that a learning algorithm is able to represent, for example all conjunctions of attribute constraints such as $\langle Sunny, Warm, ?, Strong, ?, ? \rangle$.

Version space: the subset of H that is consistent with all the training examples seen so far. It lies between the most specific boundary S and the most general boundary G. As more examples arrive, the version space shrinks.



PART–B Answers (≈250 Words Each)

Q1. Explain data manipulation operations (reading, selecting, filtering, sorting, grouping, ranking) with R code. (Expected)



Preprocessing converts raw data into a clean table. The main operations are:

Operation	Purpose	Base R	dplyr
Read	Load data into a data frame	read.csv("f.csv")	readr::read_csv()
Select	Keep chosen columns	df[, c("name","marks")]	select()
Filter	Keep rows meeting a condition	subset(df, marks > 60)	filter()
Missing values	Detect and treat NA	is.na(), na.omit()	tidyr::drop_na()
Manipulate	Create or change columns	df[["pct"]] <- df[["marks"]] / 100	mutate()
Sort	Order rows	df[order(df[["marks"]]),]	arrange()
Group	Summarise per category	aggregate(marks ~ branch, df, mean)	group_by() %>% summarise()
Rank	Position of each value	rank(-df[["marks"]])	mutate(r = rank(-marks))

Example code:

```
df <- read.csv("students.csv")
df <- df[ , c("name", "branch", "marks")]
df <- subset(df, !is.na(marks) & marks > 40)
df <- df[order(df[["marks"]], decreasing = TRUE), ]
aggregate(marks ~ branch, data = df, FUN = mean)
df[["rank"]] <- rank(-df[["marks"]], ties.method = "min")
```

Note: rank() gives tied values the average rank by default; ties.method = "min" gives them the same lowest rank.

Q2. Explain PAC learning. Find the number of training examples needed for a given epsilon and delta. (Expected)

PAC (Probably Approximately Correct) learning is a framework that asks how many examples are enough to learn a concept well. A learner is PAC if, with probability at least $1 - \delta$ (probably), it outputs a hypothesis whose true error is at most ϵ (approximately correct).

Sample complexity for a finite hypothesis space H and a learner that is consistent with the training data:

$$m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right)$$

Example: target concept is a conjunction of 10 boolean literals, so each attribute can be positive, negated or absent and $|H| = 3^{10} = 59049$. Take $\epsilon = 0.1$ and $\delta = 0.05$.

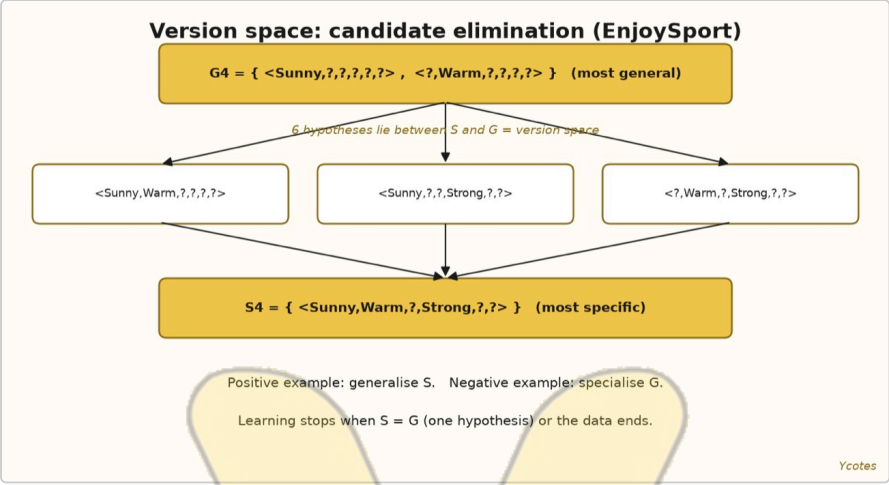
$\ln|H| = 10\ln 3 = 10.986$
$\ln(1/\delta) = \ln 20 = 2.996$
$m \geq \frac{1}{0.1}(10.986 + 2.996) = 139.8$

So **at least 140 training examples** are needed.

Interpretation: the bound grows linearly with $1/\epsilon$, logarithmically with $|H|$ and with $\ln(1/\delta)$. Smaller error, higher confidence or a richer hypothesis space needs more data.

PART–C Answers (≈500 Words Each)

Q1. Explain the candidate elimination algorithm. Trace it on the EnjoySport training data. (Expected)



Candidate elimination finds the whole **version space** by maintaining two boundaries: **S**, the most specific consistent hypotheses, and **G**, the most general consistent hypotheses.

Algorithm:

1. Initialise $G_0 = \{\langle ?, ?, ?, ?, ?, ? \rangle\}$ and $S_0 = \{\langle \phi, \phi, \phi, \phi, \phi, \phi \rangle\}$.
2. For each **positive** example d: remove from G every hypothesis inconsistent with d; for each s in S that does not cover d, replace s with its minimal generalisations that cover d (and are still below some member of G).
3. For each **negative** example d: remove from S every hypothesis that covers d; for each g in G that covers d, replace g with its minimal specialisations that exclude d (and are above some member of S).

Training data (Sky, AirTemp, Humidity, Wind, Water, Forecast, EnjoySport):

#	Sky	Temp	Humidity	Wind	Water	Forecast	Class
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Trace:

After example	S	G
0 (start)	$\langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$	$\langle ?, ?, ?, ?, ?, ? \rangle$
1 (Yes)	$\langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$	$\langle ?, ?, ?, ?, ?, ? \rangle$
2 (Yes)	$\langle Sunny, Warm, ?, Strong, Warm, Same \rangle$	unchanged
3 (No)	unchanged	$\langle Sunny, ?, ?, ?, ?, ? \rangle$, $\langle ?, Warm, ?, ?, ?, ? \rangle$, $\langle ?, ?, ?, ?, ?, Same \rangle$
4 (Yes)	$\langle Sunny, Warm, ?, Strong, ?, ? \rangle$	$\langle Sunny, ?, ?, ?, ?, ? \rangle$, $\langle ?, Warm, ?, ?, ?, ? \rangle$

Explanation of step 3: the negative example (Rainy, Cold, High, Strong, Warm, Change) is covered by the all-? hypothesis, so G is specialised using only values that S already has and that exclude Rainy, Cold or Change: Sunny, Warm and Same. Humidity = Normal is not kept because S has ? there, so it would not stay more general than S; Wind = Strong and Water = Warm are not kept because they do not exclude the negative example.

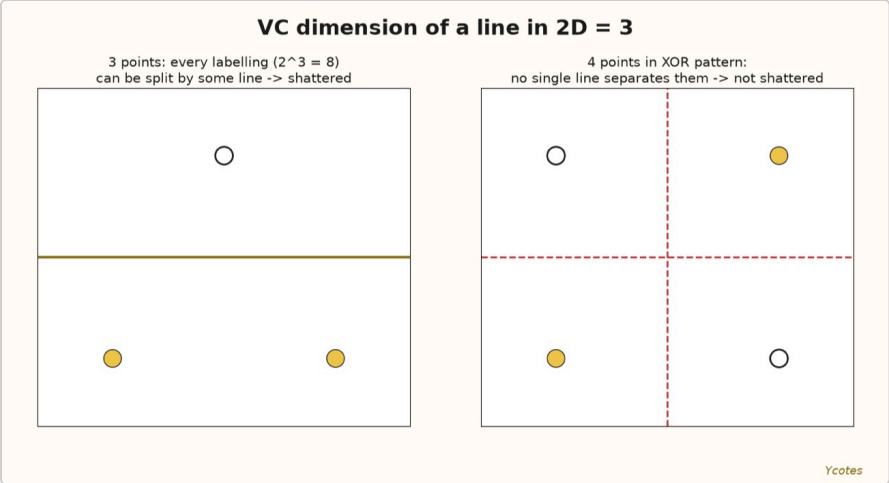
Explanation of step 4: the positive example has Forecast = Change, so $\langle ?, ?, ?, ?, ?, Same \rangle$ in G is inconsistent and removed; Water becomes ? in S.

Result: the version space has six hypotheses between $S_4 = \langle Sunny, Warm, ?, Strong, ?, ? \rangle$ and G_4: the two members of G, the S member, and three in between such as $\langle Sunny, Warm, ?, ?, ?, ? \rangle$, $\langle Sunny, ?, ?, Strong, ?, ? \rangle$ and $\langle ?, Warm, ?, Strong, ?, ? \rangle$.

Conclusion: candidate elimination returns all consistent hypotheses without enumerating H, unlike List-then-Eliminate. It is sensitive to noisy data because one wrong label can empty the version space.

Q2. Explain VC dimension and shattering. Show that the VC dimension of a line in 2D is 3. Explain hypothesis elimination (Find-S and List-then-

Eliminate). (Expected)



Shattering: a set of n points is shattered by a class H if for each of the 2^n ways of labelling the points as positive or negative there is some $h \in H$ that classifies them exactly. **VC dimension** VC(H) is the size of the largest set that H can shatter. It measures the **capacity** of a model, and it appears in generalisation bounds: more VC means more data is needed to avoid overfitting.

Line in 2D:

- **VC ≥ 3:** take 3 points that are not collinear. There are $2^3 = 8$ labellings: all same (a line far to one side), and for the other six, one point is alone on one side of a line that separates it from the other two. All 8 can be separated, so 3 points are shattered.
- **VC < 4:** for any 4 points, consider the XOR labelling where diagonal points share a label. No single straight line can separate the two positives from the two negatives. If the 4 points form a triangle with one inside, the inner point cannot be separated from the outer three either. So no 4 points can be shattered.
- Therefore **VC dimension = 3**. In general, a hyperplane in d dimensions has VC dimension $d + 1$.

Hypothesis elimination:

Algorithm	Working	Output	Limitation
Find-S	Start with the most specific hypothesis; for each **positive** example generalise it minimally	One most specific hypothesis	Ignores negative examples, cannot tell if other hypotheses fit
List-then-Eliminate	List all hypotheses in H; remove every one inconsistent with any example	The complete version space	Needs enumeration of H, impractical for large H

Find-S on the EnjoySport positives (1, 2, 4): start $\langle \phi, \ldots, \phi \rangle$, after 1 gives $\langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$, after 2 gives $\langle Sunny, Warm, ?, Strong, Warm, Same \rangle$, after 4 gives $\langle Sunny, Warm, ?, Strong, ?, ? \rangle$.

Link: candidate elimination removes the weakness of both by tracking only the boundaries S and G.

