Nithish S
B.Tech. AI & DS
953621243039@ritrjpm.ac.in

Unit - 1

1.0 Linear Algebra for Machine Learning

- Mathematical Foundation that solves problem of representing data.
- It is the math of arrays → {Vectors, Matrices and Tensors}

Vector

$$x = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}$$

Matrix → 2D - Array

$$x = \begin{bmatrix} a_{11} & a_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Tensor → Matrix with ndim >= 3

$$x = \begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

Applications of Linear Algebra

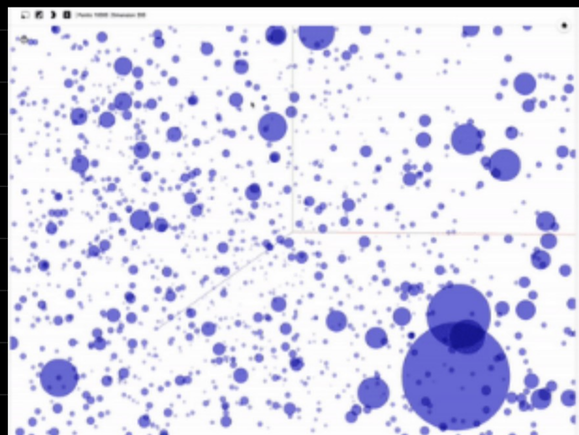
- Data Representation (Vectors, Matrices and Tensors).
- NLP → (converting words to vectors) {Word2Vec}
- Dimensionality Reduction - (eigen value, eigen vector for large dimensional data) {PCA}

i) Data Representation

- The DATA being fed to ML models must be converted into arrays.
- The computations performed on these array include operations like (dot product).
- The output is also represented as transformed matrix / tensor of numbers.

ii) Word - Embeddings

- Representing Large dimensional data into smaller dimensional vector.



- Mostly used in NLP, and each word can be embedded as vectors.
- By using vectors it will be efficient for representation of the words.

iii) Principal Component Analysis

- Eigen Vectors are used in PCA.
- Allows us to reduce dimension of the data while keeping the essence of all of them.

Data to Vectors

- Vectors are basically 1-D array of numbers but they are said to have both magnitude and direction

Weight	Age	Height
80	20	180
75	22	187
82	19	175

$$\longrightarrow \begin{bmatrix} 80 \\ 20 \\ 180 \end{bmatrix} \longrightarrow$$

1.1 Intro to ML

- Teaching a computer / mathematical model by feeding/showing data and predicting a new value with respect to the past data.
- The predicting would just be a probability value and truness of the prediction would depend on quality of data that was fed to the model.

Machine Learning

- Supervised
  - Regression
  - Classification
- Un-Supervised
  - Clustering
  - Dimensionality Reduction

Supervised $\longrightarrow$ Labelled Data.

Un-Supervised $\longrightarrow$ No Labelled Data is present.

Important Components (supervised)

- Dependent Features
- In-Dependent Features

Age	Weight
24	62
25	74
27	67

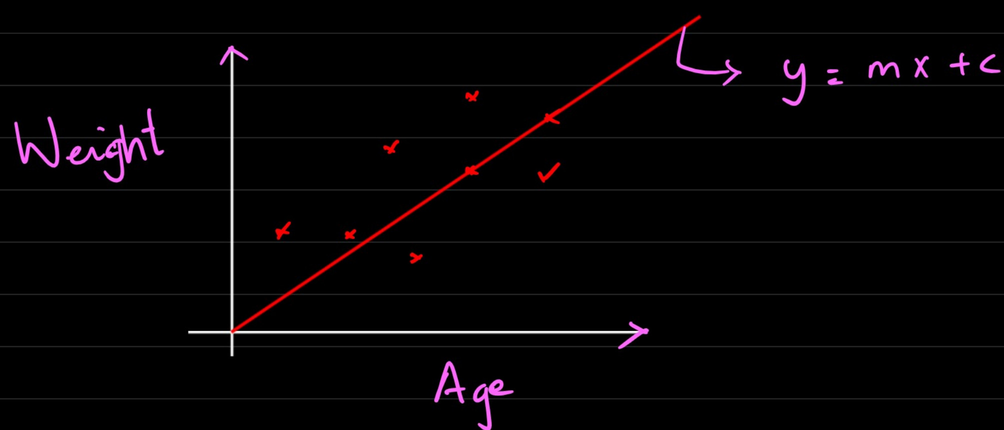
Age → $\boxed{\text{hypothesis}}$ → o/p weight

Supervised - ML

⇓

i) Regression Problem Statement

⇒ If the output is a Continuous Variable then we call it a regression problem



Predict 'X' with respect to 'Y'.

ii) Classification Problem Statement

playing	sleeping	studing	Pass / Fail
2	6	3	P
2	8	2	F
1	5	4	P

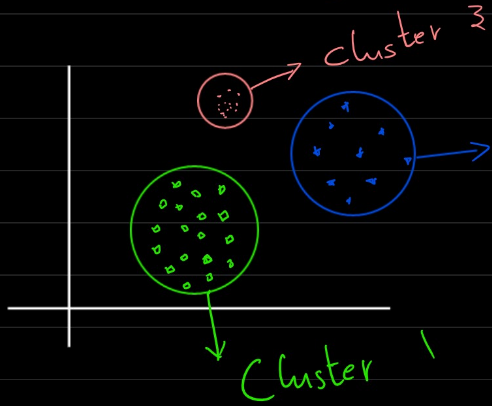
→ Whenever output has fixed number of categories it is called classification problem.

Unsupervised - ML

i) Clustering ⟶ "GROUPING" {No Dependent Feature}

∘) Identifying group of similar objects in data.

Ex: Customer Segmentation



"CLUSTERING IS NOT CLASSIFICATION, CLUSTERING HAS NO o/p FEATURE"

ii) Dimensionality Reduction

∘) Suppose there are 1000 features,

∘) Converting them into 100 features is called dimensionality Reduction. Ex - PCA, LDA

1.3 Examples of ML applications

∘) Weather prediction app - using Linear / Logistic Regression.

∘) Sentiment Analysis - using clustering.

Classification vs Clustering

Classification	Clustering
i) Uses Labeled data.	Uses unlabeled data
ii) There is a need for training & testing dataset.	No need to train & test dataset.
iii) Complex compared to Clustering.	Less complex compared to classification
iv) Ex: Logistic Regression, Naive Bayes, SVM.	K-Means, C-Means.

1.4 Vapnik - Chervonenkis Dimension

- Measure of complexity / capacity of ML model.
- Represents maximum number of points that can be seperated into two models with perfect accuracy by a given model.
- Tells us if model is overfitting or underfitting the given data.

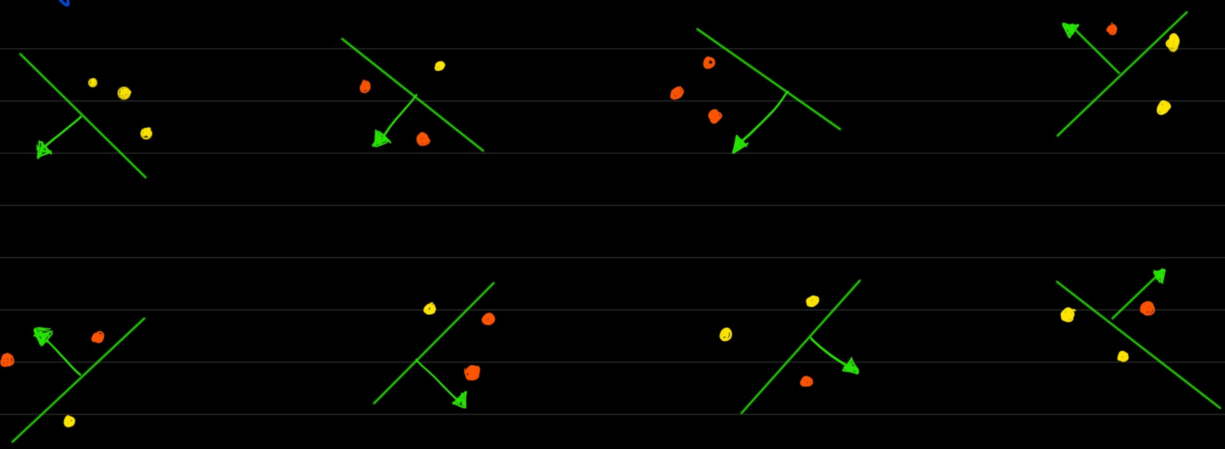
- Also helps us in choosing a correct classification model.

- "The cardinality of the Largest set of points that Classification algorithm can shatter.

⇒ "Shatter" → Given a set of points, a model shatters this set if it can label all possible combinations of positive and negative points in that set.

⇒ VC Dimension for Linear Classifier is {Atleast 3}, since it can shatter this configuration of 3 points.

⇒ In each of $2^3 = 8$ possible assignment of positive and Negative, the classifier is able to seperate the classes.



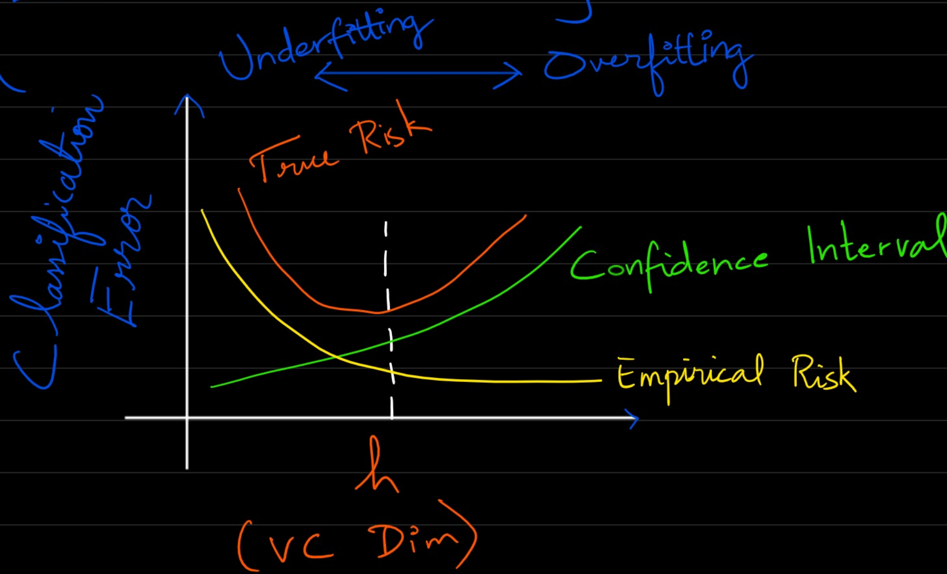
⇒ If the configuration has 4 points, we might need to draw 2 lines to seperate them.

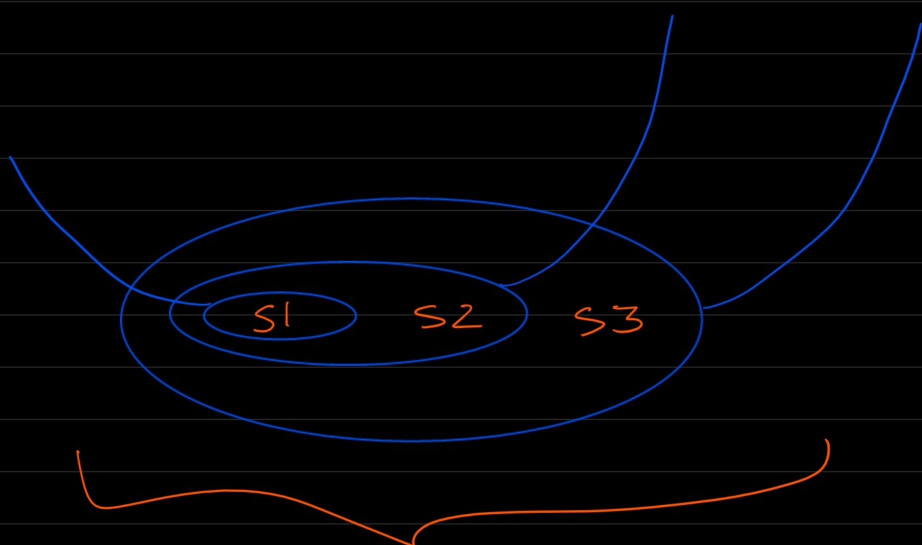
⇒ But, "VC Dimension is exactly 3", not greater than 3.

In Simple → "Measures how good the ML model is understanding patterns in data."

Applications of VC Dimension

- Used in classifying algorithms based on their complexities.
- The complexity of a classification algorithm, which is directly related to VC dimension, is related to {bias - variance trade-off}.





Effects of model complexity.

'S_n' represents set of models similar in VC dimension.

1.5 Probably Approximately Correct (PAC) Learning

- Framework used for mathematical analysis.
- A PAC Learner tries to learn a concept by selecting a hypothesis from a set of hypothesis which has a low generalization error.

Explained to a kid

- Imagine there are a bag of candies and you want to find out which is your favorite, You can do this by tasting each candy and choosing your best.
- Like this, the computer tries different solutions and returns the best among them.

{ Goal is to find a solution that is probably approximately correct meaning the solution is likely to be correct and not guarenteed to be correct }

PAC in a mathematical aspect

- For a given class 'c', examples are drawn from some random but fixed probability distribution 'p(x)'.

- We have to find number of examples 'N', with probability atleast $(1-\delta)$ a system can learn a concept with error at most 'ε'.

$$P(c \Delta h \leq \varepsilon) \geq 1-\delta$$

- $c \Delta h \rightarrow$ region of difference between 'c' and 'h'.

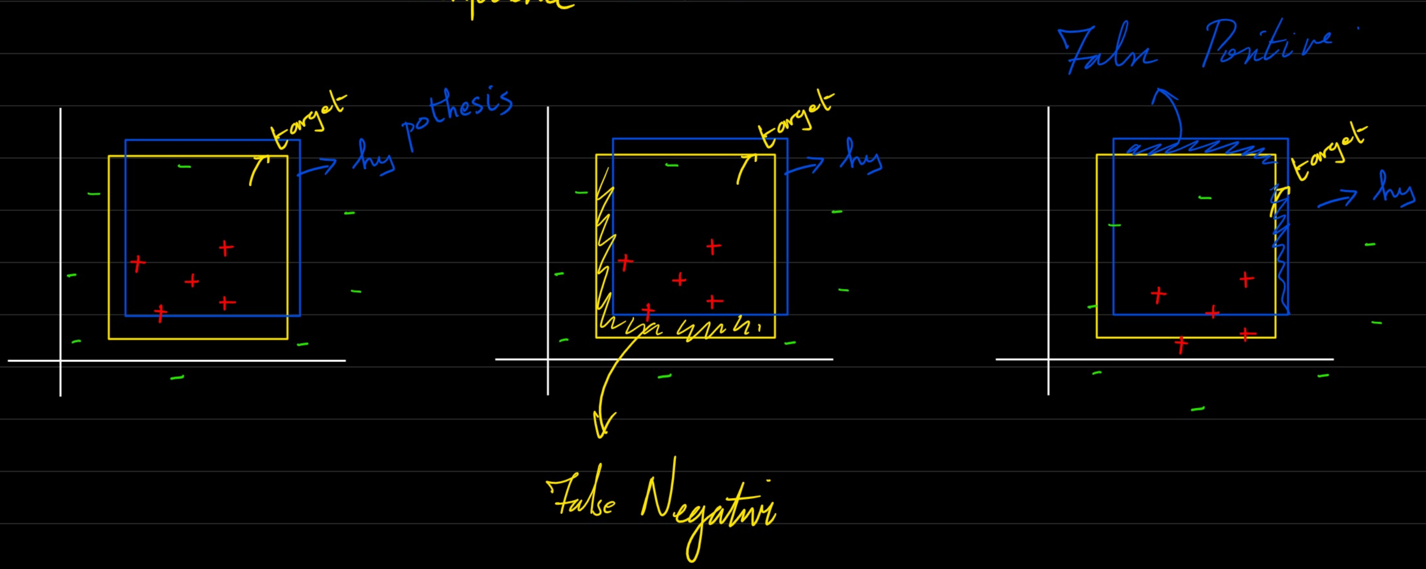
  $c \rightarrow$ class, $h \rightarrow$ hypothesis.

- $\varepsilon \rightarrow$ gives an upper bound on the error.

  Accuracy $= 1-\varepsilon$.

- $\delta \rightarrow$ Probability of failure in achieving accuracy.

  Confidence $= 1-\delta$



- A hypothesis is said to be correct, if the error is $\leq$ 'ε'.

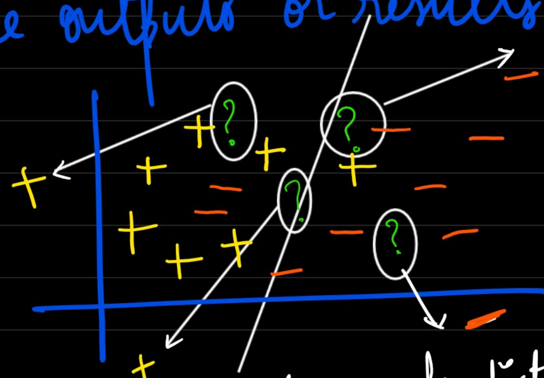
$$\text{error}_D(h) = \Pr_{x \in D}\left(c(x) \neq h(x)\right)$$

1.6 Hypothesis Space

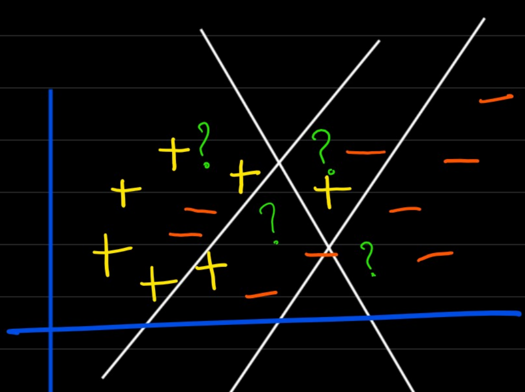
- It is a set of all possible functions that a learning algorithm can use to approximate the target function in PAC learning problem.

- The choice of hypothesis space is important because it determines the type of function that an the algorithm can consider as solution.

Ex: Suppose we have a test data for which we have to determine the outputs or results.



we can predict by drawing a co-ordinate.



We could have drawn coordinates anywhere, All these possible ways in which we can divide the co-ordinates is called "HYPOTHESIS SPACE".

1.7 Inductive Bias

There is a model, and it is given a data and an learning algorithm. The set of assumptions that the learner uses to predict outputs of the given input that it has not encountered.

1.8 Generalization

The ability of the model to adopt properly to new / unseen data, drawn from the same distribution as the one used to create the model.

1.9 Bias - Variance trade - off

Bias and Variance are two types of errors that can occur in a model's prediction.

Bias → Difference between actual and predicted values

A model with high bias tends to underfit, meaning it is too each to find the patterns in the data.

Variance → Variability of models predictions on different datas. A model with high variance will be overfitting. Also, it might perform well with the training data but it might be wrong with new data.

Since, it is overfitting the model would try to capture unwanted noise in the dataset.

Bias - Variance tradeoff

Ability of a model to balance bias and Variance.

If the model has the following,

- high bias, Low variance → Underfit
- low bias, high variance → Overfit

"GOAL IS TO FIND OPTIMAL BALANCE FOR BOTH."

Find-S Algorithm {with example dataset}

Ball	Color	Size
1	Red	Big
2	Red	Small
3	Blue	Big
4	Blue	Small
5	Green	Big
6	Green	Small
7	Red	Big
8	Blue	Small
9	Green	Big
10	Red	Small

Steps in Find-S

1. Start with empty hypothesis.
2. Observe & update the hypothesis.
3. If matches make no change.
4. Repeat Step 2 & 3.

1. From the above dataset we are going to observe a girl play and. we will arrive at a conclusion, (whether) the girl will play with a specific ball or not.

2. Our first step → empty hypothesis < >

3. Girl picks first ball → < Red, Big > → girl will play.

4. Picks the Second ball → < Red > → girl will play.

5. Picks 3rd balls it is found to be blue and big

then hypothesis becomes → < Red, Blue, Big > → girl plays.

6. Picks 4th ball it is found to be blue but small so we again update the hypothesis → < Red, Blue > → girl plays.

⋮

7. Girl will play if the ball is < Red, Blue, Green >

Candidate - Elimination algorithm

Example	Citations	Size	InLibrary	Price	Editions	Buy
1	Some	Small	No	Affordable	One	No
2	Many	Big	No	Expensive	Many	Yes
3	Many	Medium	No	Expensive	Few	Yes
4	Many	Small	No	Affordable	Many	Yes

S_0 : (0,0,0,0,0)

S_1 : (0,0,0,0,0)

S_2 : (Many, big, No, Exp, Many) (2)

S_3 : (Many, ?, No, Exp, ?) (3)

S_4 : (Many, ?, No, ?, ?) (4)

G_0 : (? ? ? ? ?)

G_1 : (Many ? ? ? ?), (? Big ? ? ?), (? Medium ? ? ?), (? ? ? Exp ?), (? ? ? ? Many), (? ? ? ? Few).

G_2 : (Many ? ? ? ?), (? Big ? ? ?), (? ? ? Exp ?), (? ? ? ? Many)

G_3 : (Many ? ? ? ?), (? ? ? Exp ?)

G_4 : (Many ? ? ? ?)