Concept Learning

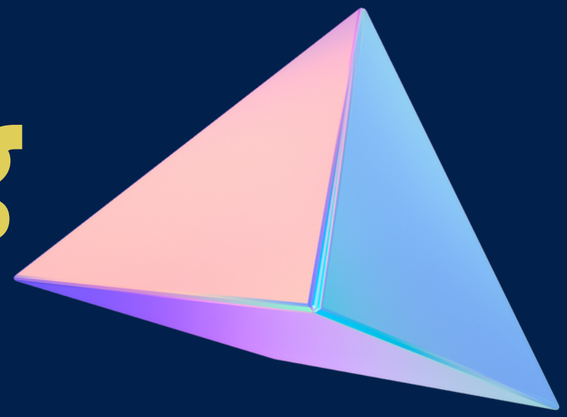
UNIT – 1

INTRODUCTION TO MACHINE LEARNING

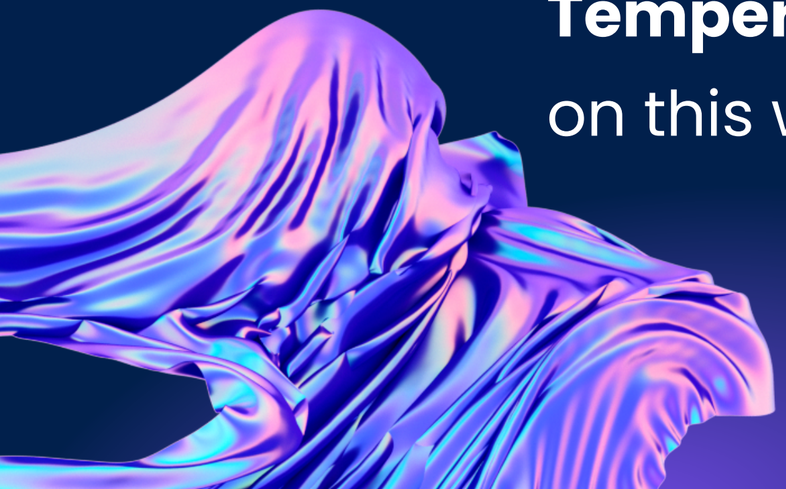
Review of Linear Algebra for machine learning; Introduction and motivation for machine learning; Examples of machine learning applications, Vapnik-Chervonenkis (VC) dimension, Probably Approximately Correct (PAC) learning, Hypothesis spaces, Inductive bias, Generalization, Bias variance trade-off.

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
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

Concept Learning



- Taking a very simple example, one possible target concept may be to **Find the day when my friend Ramesh enjoys his favorite sport**.
- We have some attributes/features of the day like, **Sky, Air Temperature, Humidity, Wind, Water, Forecast** and based on this we have a target Concept named **EnjoySport**.



Design the problem formally with TPE(Task, Performance, Experience):

- **Problem:** Leaning the day when Ramesh enjoys the sport.
- **Task T:** Learn to predict the value of EnjoySport for an arbitrary day, based on the values of the attributes of the day.
- **Performance measure P:** Total percent of days (EnjoySport) correctly predicted.
- **Training experience E:** A set of days with given labels (EnjoySport: Yes/No)

Hypothesis Representation

hi(x) := <x1, x2, x3, x4, x5, x6>

where x1, x2, x3, x4, x5, and x6 are the values of Sky, AirTemp, Humidity, Wind, Water, and Forecast.

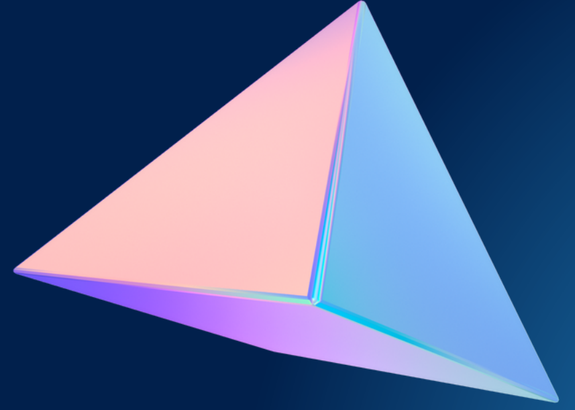
- The two most popular approaches to find a suitable hypothesis, are:

1. **Find-S Algorithm**
2. **List-Then-Eliminate Algorithm**

Find - S Algorithm

- The Find-S algorithm is a basic concept learning algorithm in machine learning.
- It finds the most specific hypothesis that fits all positive examples.
- This algorithm considers only the positive examples.
- The find-S algorithm starts with the most specific hypothesis and generalizes this hypothesis each time it fails to classify an observed positive training data.
- Hence, the Find-S algorithm moves from the most specific hypothesis to the most general hypothesis.

Hypothesis Representation



- ? – indicates any value is acceptable
- Φ – indicates no value is acceptable
- Specify a single value for an attribute (Eg: Green)
- General hypothesis is represented by {?, ?, ?, ?, ?}
- Specific hypothesis is represented by {Φ, Φ, Φ, Φ}

Steps in Find - S Algorithm

- Start with the most specific hypothesis.
- h = {ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}
- Take a particular example and if it is negative, then no changes occur to the hypothesis.
- If the example is positive and we find that our initial hypothesis is too specific then we update our current hypothesis to a general condition.
- Keep repeating the above steps till all the training examples are complete.
- After we have completed all the training examples we will have the final hypothesis when can use to classify the new examples.

Example	Colour	Toughness	Fungus	Appearance	Poisonous
1	Green	Hard	No	Wrinkled	Yes
2	Green	Hard	Yes	Smooth	No
3	Brown	Soft	No	Wrinkled	No
4	Orange	Hard	No	Wrinkled	Yes
5	Green	Soft	Yes	Smooth	Yes

- First, we consider the hypothesis to be a more specific hypothesis.
- Hence, our hypothesis would be :

h = {ϕ, ϕ, ϕ, ϕ}

- The data in 1st sample is **{GREEN, HARD, NO, WRINKLED }**.
- We see that our initial hypothesis is more specific and we have to generalize it.
- Now, the hypothesis becomes :

h = { GREEN, HARD, NO, WRINKLED }

- Consider sample 2, which has a negative outcome.
- Hence we neglect this example and our hypothesis remains the same.

h = { GREEN, HARD, NO, WRINKLED }

- Consider sample 3, which has a negative outcome. Again we neglect this example and our hypothesis remains the same.

h = { GREEN, HARD, NO, WRINKLED }

- Consider sample 4
- The data present in example 4 is **{ORANGE, HARD, NO, WRINKLED }.**
- We compare every single attribute with the initial data and if any mismatch is found we replace that particular attribute with a general case (” ? ”).
- After doing the process the hypothesis now becomes :

h = { ?, HARD, NO, WRINKLED }

- Consider sample 5 :
- The data present in example 5 is **{ GREEN, SOFT, YES, SMOOTH }**. We compare every single attribute with the initial data and if any mismatch is found we replace that particular attribute with a general case (” ? ”). After doing the process the hypothesis becomes :

h = { ?, ?, ?, ? }

Hence, for the given data the final hypothesis would be :

Final Hypothesis: h = { ?, ?, ?, ? }

Candidate Elimination Algorithm

- Candidate Elimination algorithm is used to find consistent hypothesis for the given set of training examples.
- It considers both positive and negative samples.

Sky	Temperature	Humid	Wind	Water	Forest	Output
sunny	warm	normal	strong	warm	same	yes
sunny	warm	high	strong	warm	same	yes
rainy	cold	high	strong	warm	change	no
sunny	warm	high	strong	cool	change	yes

Initialize, **S0 = {Φ, Φ, Φ, Φ, Φ, Φ} (Specific boundary)**

G0 = {?,?,?,?,?,?} (General Boundary)

- For instance 1: **<'sunny', 'warm', 'normal', 'strong', 'warm ','same'>** and positive output.

G1 = G0

S1 = ['sunny', 'warm', 'normal', 'strong', 'warm ','same']

All question marks match with the example. So replace null values with the attribute values

- For instance 2: **<'sunny', 'warm', 'high', 'strong', 'warm ','same'>** and positive output.

G2 = G0

S2 = ['sunny', 'warm', ?, 'strong', 'warm ', 'same']

- For instance 3: **<'rainy', 'cold', 'high', 'strong', 'warm ', 'change'>** and negative output.

G3 = [['sunny', ?, ?, ?, ?, ?], [?, 'warm', ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, ?], [?, ?, ?, ?, ?, 'same']]

- S3 = S2
- For instance 4 : <'sunny', 'warm', 'high', 'strong', 'cool', 'change'> and positive output.
- G4 = G3
- S4 = ['sunny', 'warm', ?, 'strong', ?, ?]

S

⟨Sunny, Warm, ?, Strong, ?, ?⟩

⟨Sunny, ?, ?, Strong, ?, ?⟩

⟨Sunny, Warm, ?, ?, ?, ?⟩

⟨?, Warm, ?, Strong, ?, ?⟩

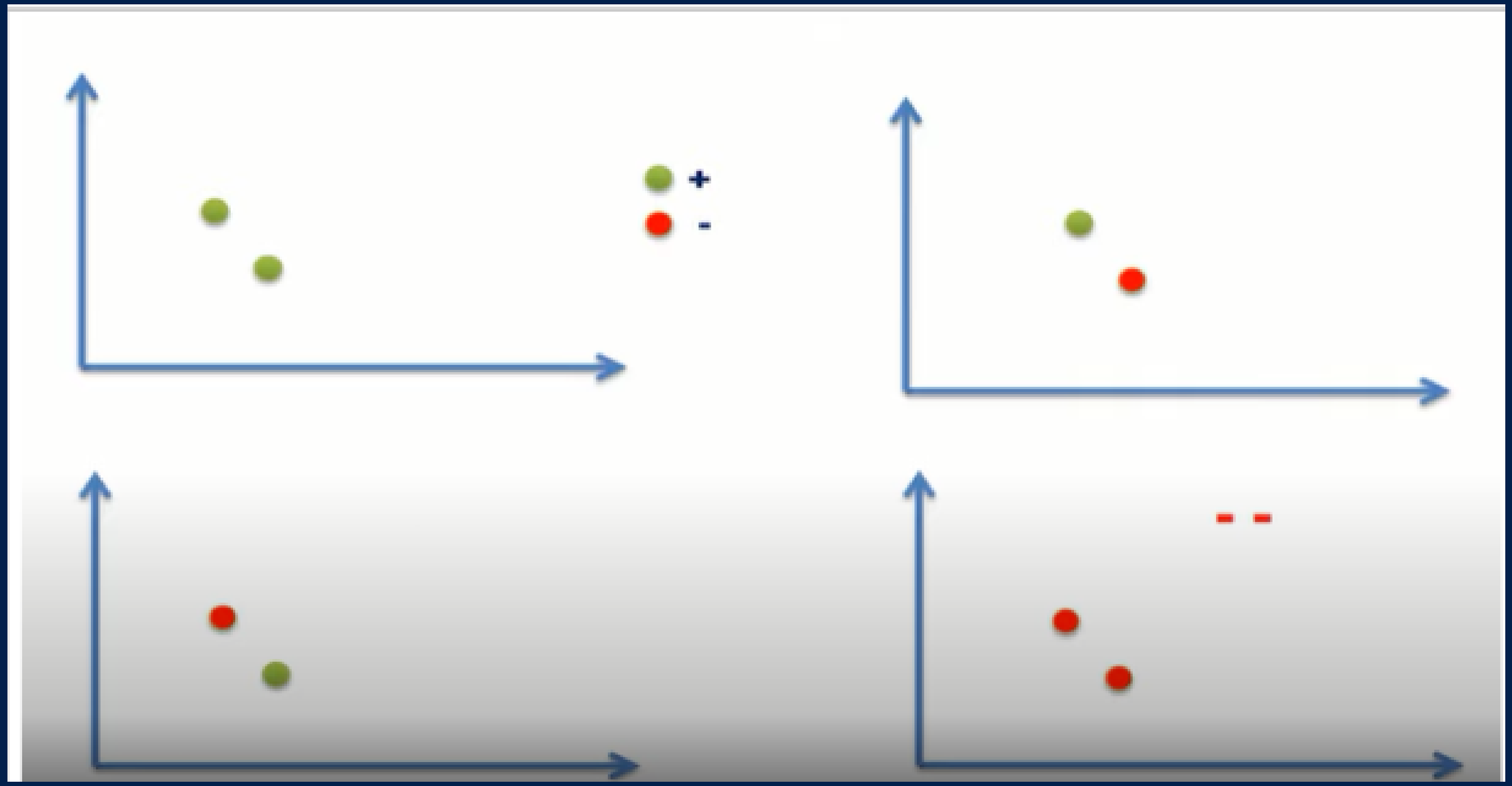
G

⟨Sunny, ?, ?, ?, ?, ?⟩

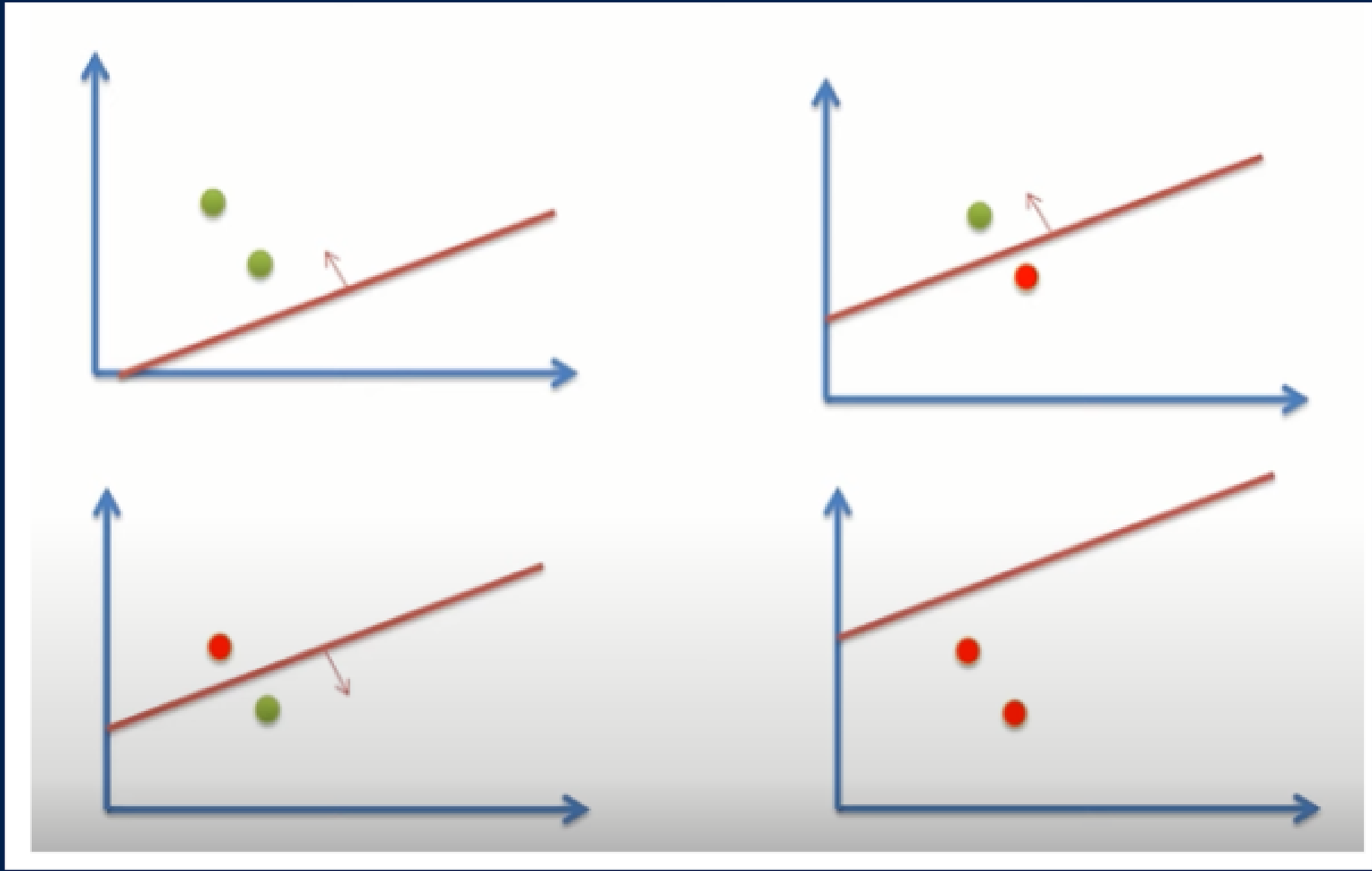
⟨?, Warm, ?, ?, ?, ?⟩

VAPNIK CHERVONENKIS (VC) DIMENSION

Linear Classifier with two data points

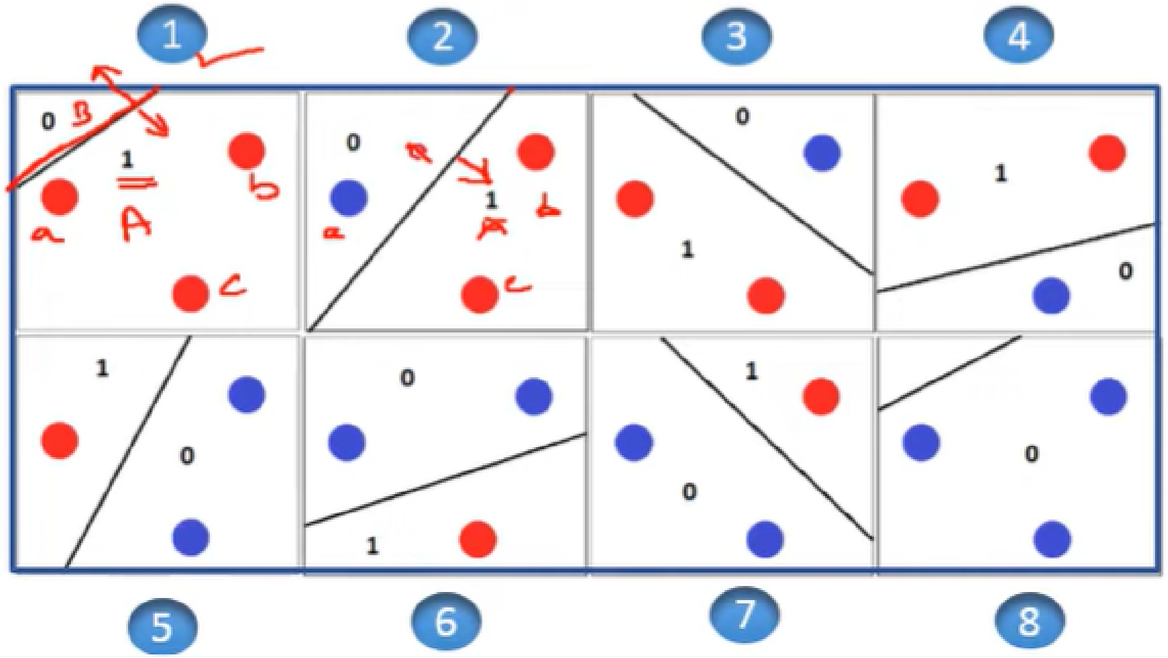


Linear Classifier with two data points



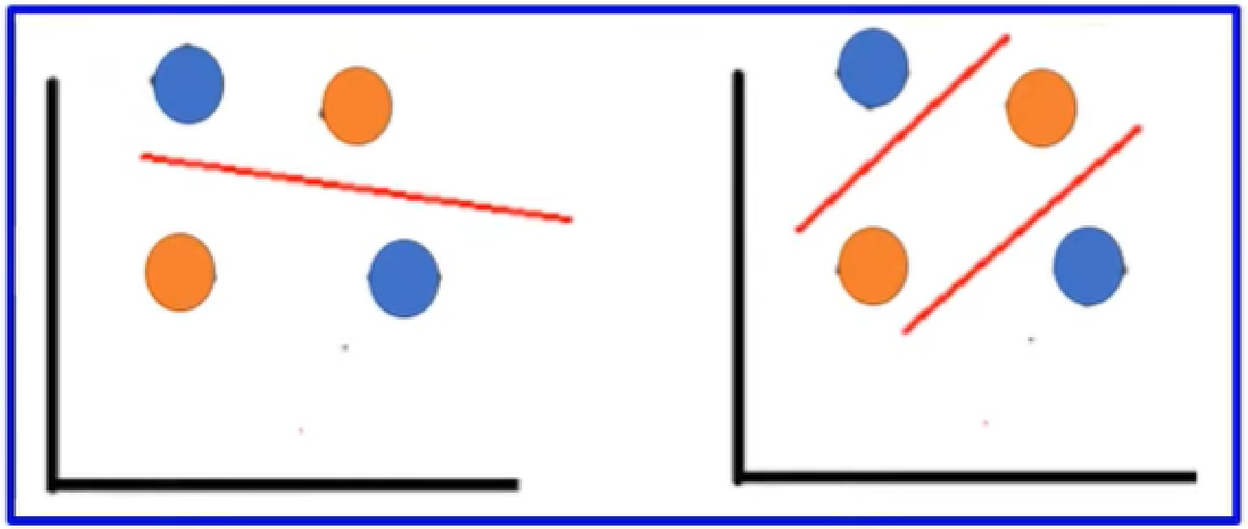
Linear Classifier with three data points

- Binary classification with three data points (in 2D space)
- The 3 points can take either class A (+) or class B (-) which gives us 2^3 (=8) possible combinations (or learning problems).
- a line can shatter 3 points (in general position).



Linear Classifier with four data points

- Now, for the case of 4 points, we can have maximum of 2^4 (=16) possible combinations.
- In Figure that the line was **unable to shatter the two classes**.
- So, we can say that the linear classifier can shatter at most 3 points.



Vapnik Chervonenkis (VC) dimesion

- A dataset contains N points.
- These N points can be labeled in 2^N ways as positive & Negative.
- Important terminology in VC dimension is Shattering.
- Shattering is the ability of the model to classify a set of points perfectly.
- A hypothesis h belongs to H that separates the positive samples from the negative, then we say that H shatters N points.

Shattering

A set of N points is said to be shattered by a hypothesis space H , if there are hypothesis(h) in H that separates positive examples from the negative examples in all of the 2^N possible ways

Vapnik Chervonenkis (VC) dimesion

- The maximum number of points that can be shattered by H is called VC dimension.
- Vapnik Chervonenkis (VC) dimension is the measure to compute the capacity of classification algorithm.

- The VC dimension VC (H), of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H.
- If arbitrarily large finite sets of X can be shattered by H, then VC (H) ≡∞

Probably Approximately Correct (PAC Learning)

PAC – Mathematical analysis of Machine Learning

Goal of PAC -> With high probability (“probably”), the selected hypothesis with have low error (Approximately correct”)

In PAC, we have two small parameters, ε and δ.

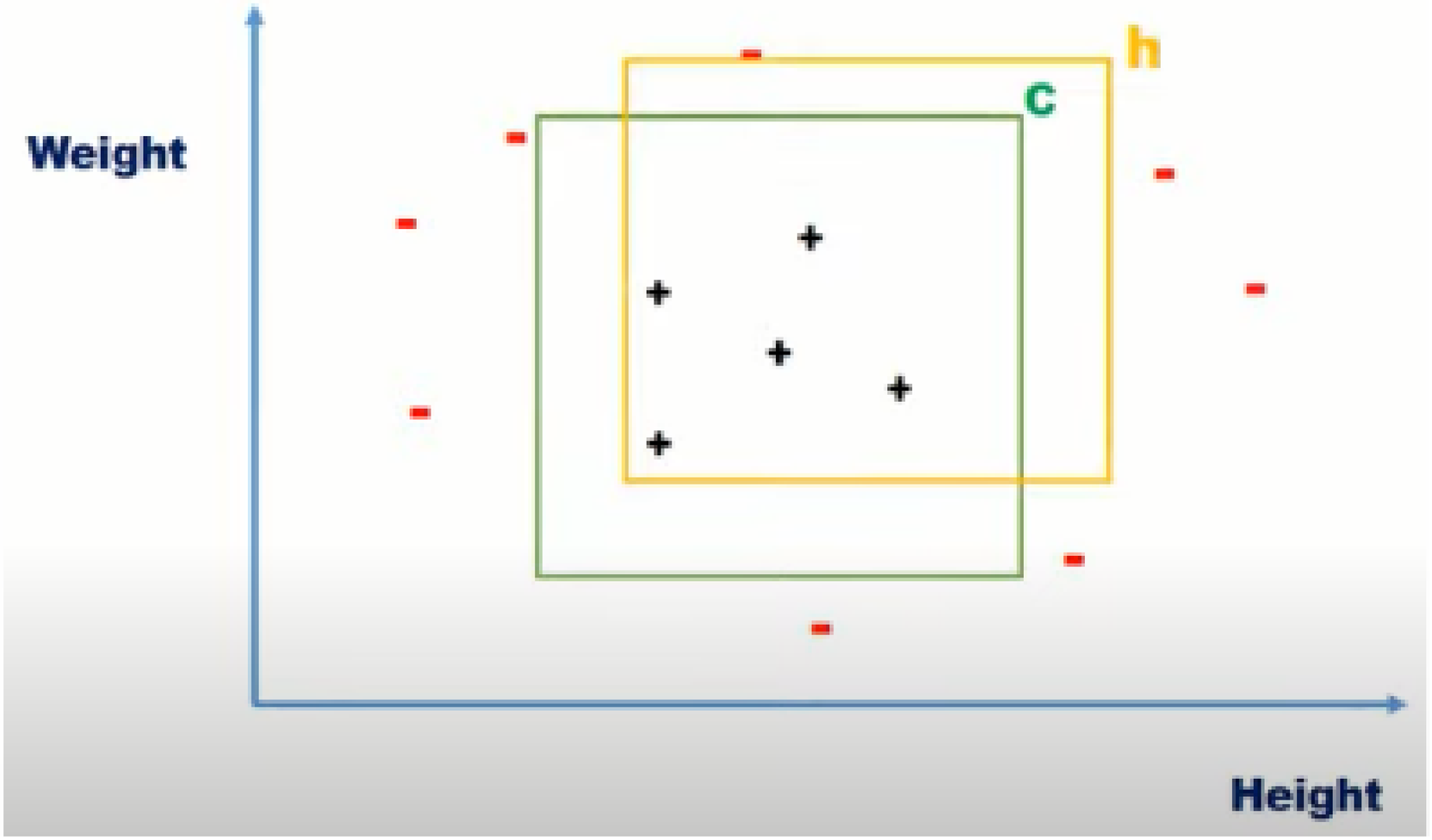
ε gives an upper bound on the error

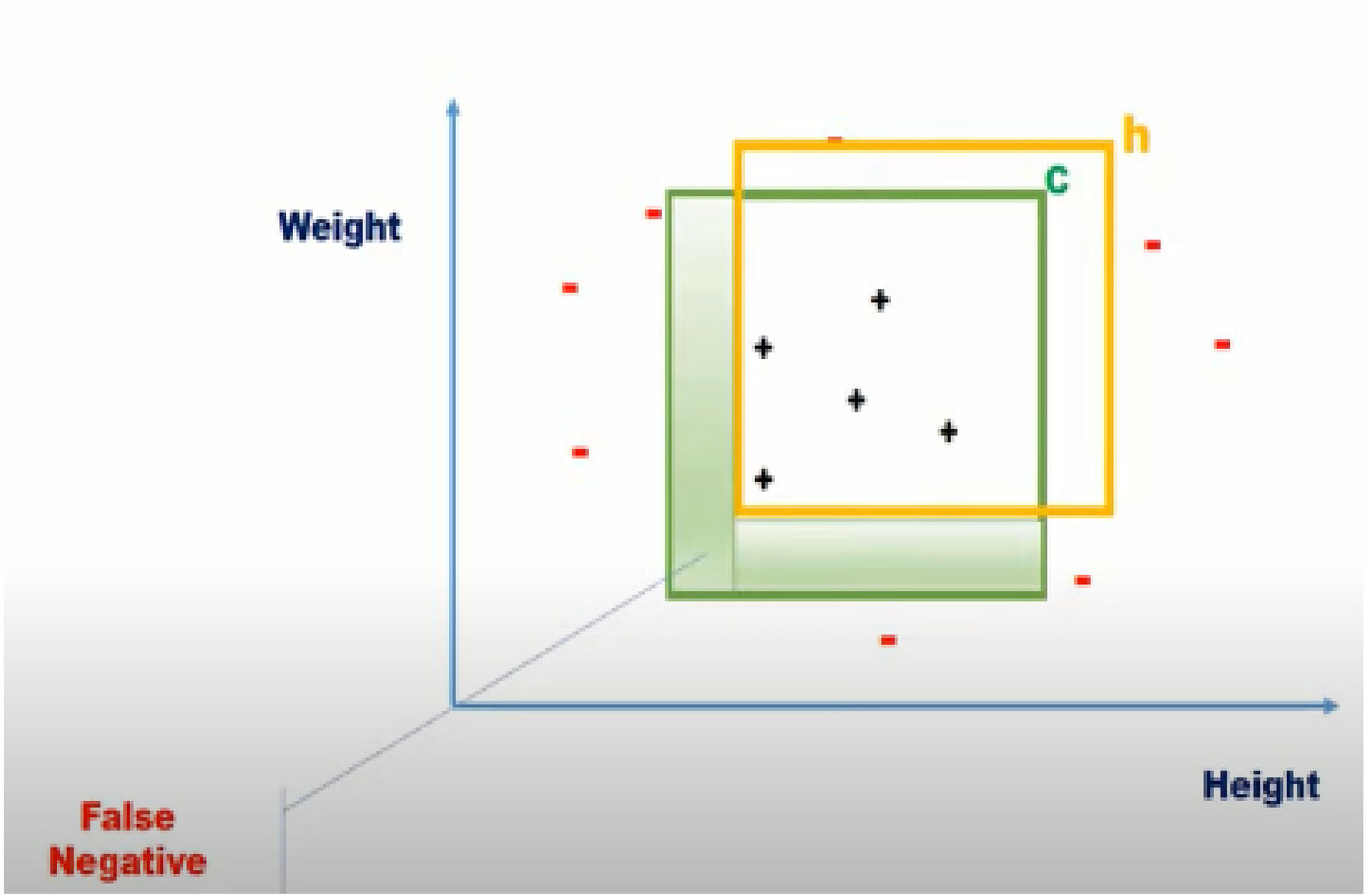
Accuracy = 1- ε

δ gives the probability of failure in achieving this accuracy.

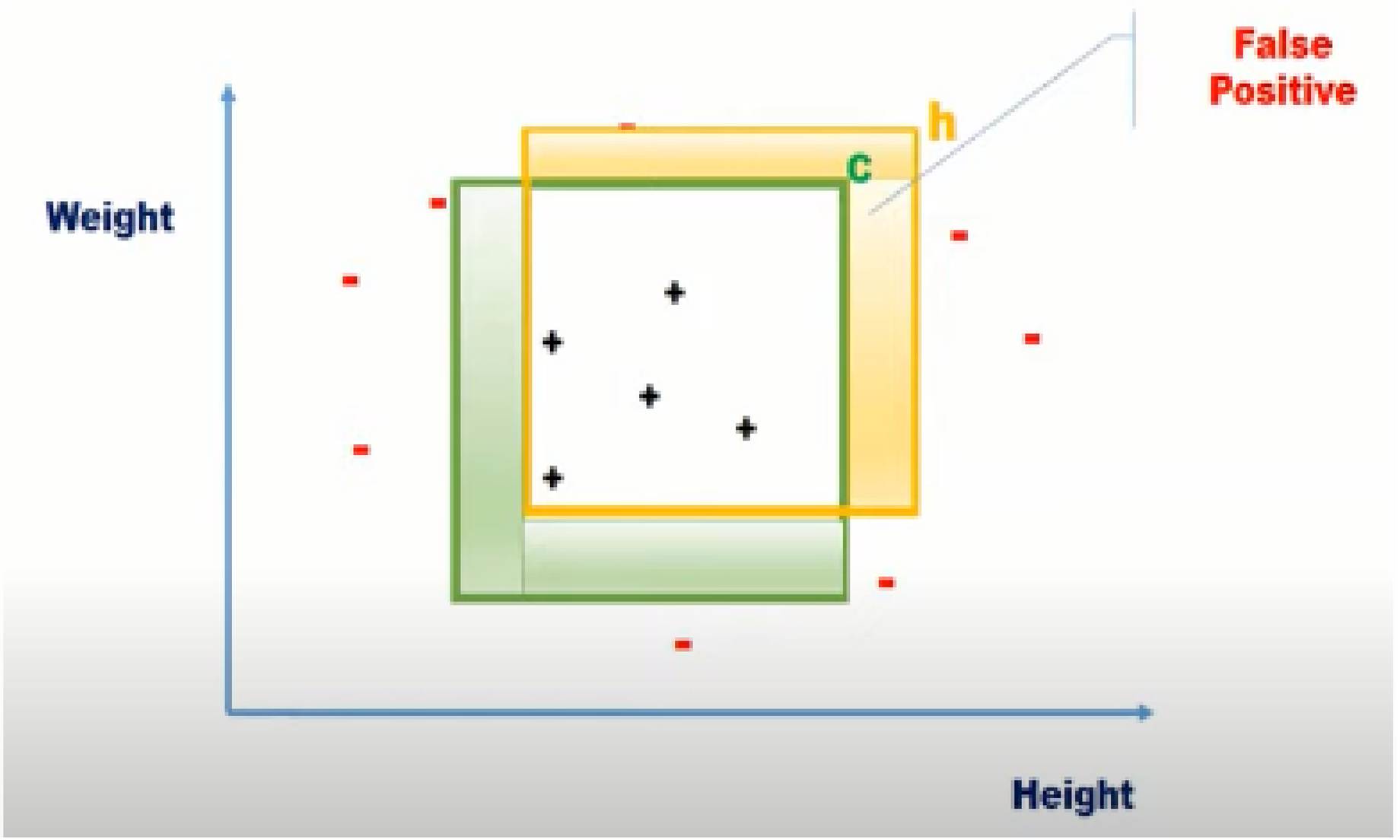
Confidence = 1- δ

c -> Target function
h -> Hypothesis




h
c
Weight
Height
False
Negative

Error Region : c XOR h



Approximately Correct

A hypothesis is said to be approximately correct , if the error is less than or equal to ε, where $0 \leq \varepsilon \leq 1/2$

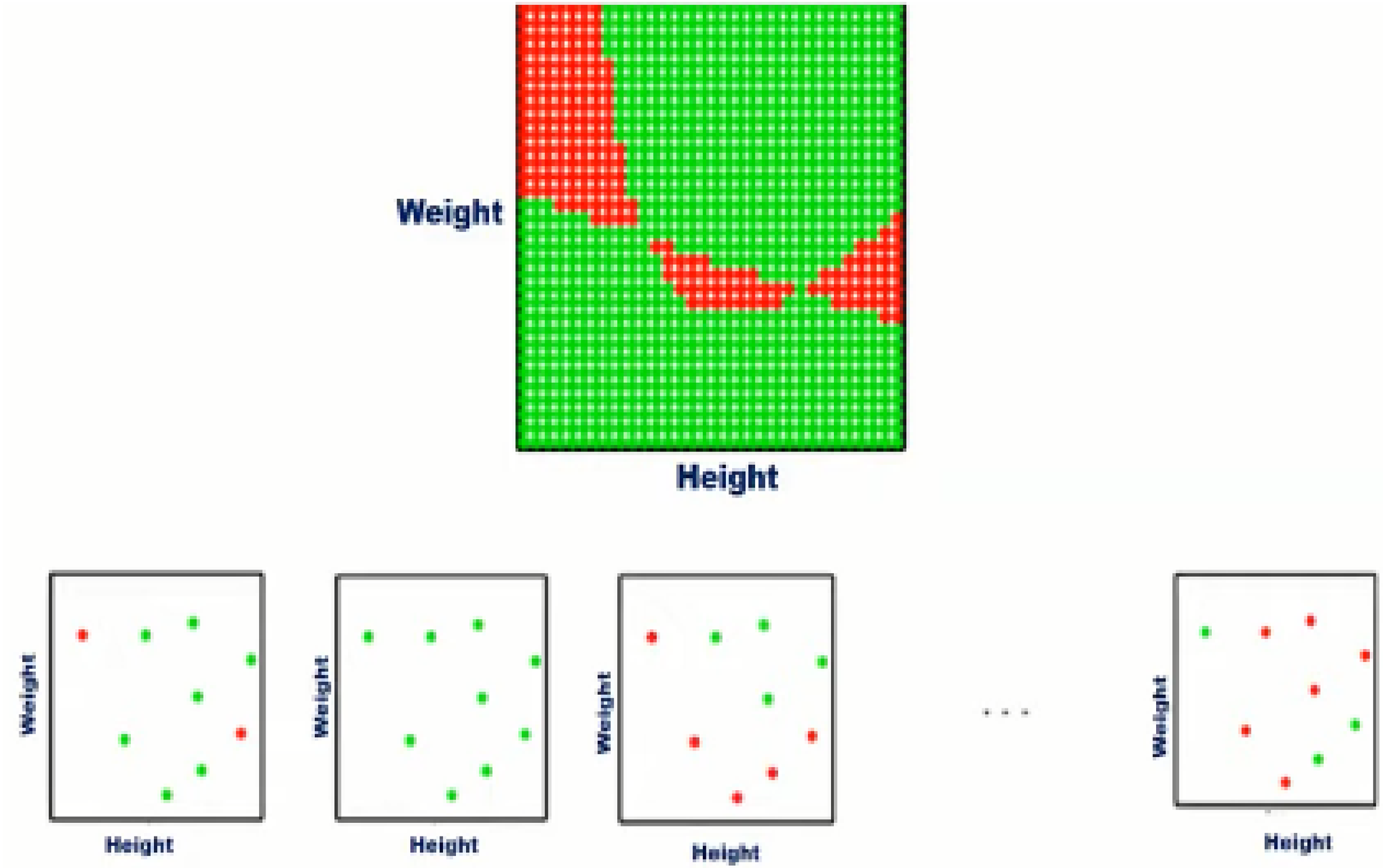
i.e., $P(C \oplus h) \leq \varepsilon$

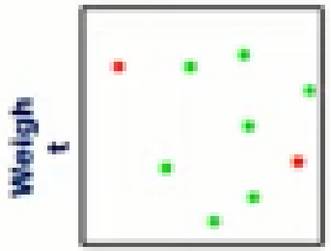
Probably Approximately Correct

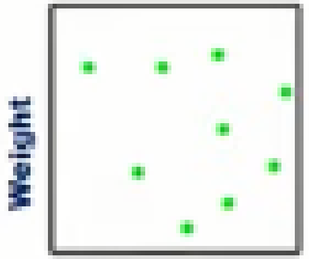
The goal is to achieve low generalization error with high probability.

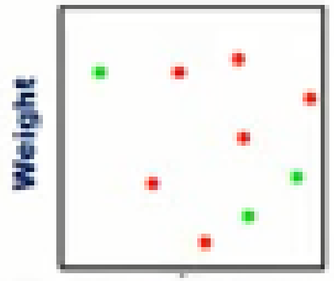
$$Pr(Error(h) \leq \varepsilon) > 1 - \delta$$

$$\text{i.e. } Pr(P(C \oplus h) \leq \varepsilon) > 1 - \delta$$


Weight
Height
Weight
Height
Weight
Height
Weight
Height
. . .
Weight
Height







Error(H1)

0.04

0.03

0.09

0.06

0.025

0.049

0.04

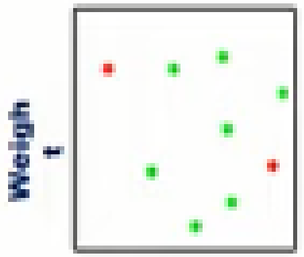
0.03

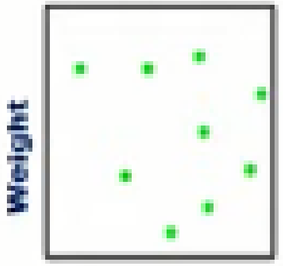
0.05

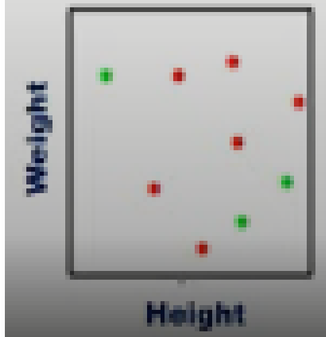
0.043

$\varepsilon = 0.05$

$\delta = 0.20$







Error(H1)

0.04

0.03

0.09

0.06

0.025

0.049

0.04

0.03

0.05

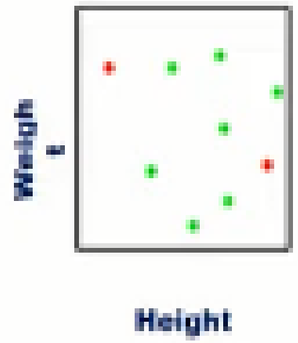
0.043

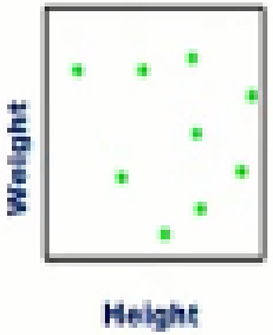
ε= 0.05 δ =0.20

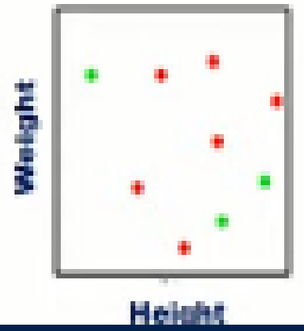
P (H1) = 8/10=0.80

P (H1) = 8/10=0.80 >=1- 0.20

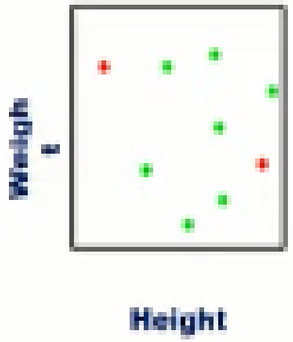
Hence, H1 is probably Approximately Correct

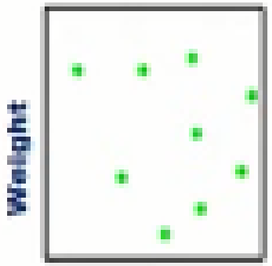


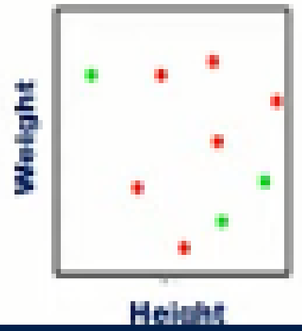




Error(H1)	Error(h2)
0.04	0.04
0.03	0.035
0.09	0.039
0.06	0.06
0.025	0.025
0.049	0.059
0.04	0.04
0.03	0.03
0.05	0.55
0.043	0.043







Error(H1)	Error(h2)
0.04	0.04
0.03	0.035
0.09	0.039
0.06	0.06
0.025	0.025
0.049	0.059
0.04	0.04
0.03	0.03
0.05	0.55
0.043	0.043

$\varepsilon = 0.05$ $\delta = 0.20$

P (H1) = 8/10=0.80

P (H1) = 8/10=0.80 >=1- 0.20

Hence H1 is probably approximately correct

P(H2)= 7/10=0.70

P(H2)= 7/10=0.70 <1-0.20

Hence H2 is not probably approximately correct

Formal Definition of PAC Learning

Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.

C is PAC-learnable by L using H if for all $c \in C$, distributions D over X, ε such that $0 < \varepsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner L will output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \varepsilon$ with probability at least $(1 - \delta)$, in time that is polynomial in $1 / \varepsilon$, $1 / \delta$, n and size(c)

Prob[errD(h)<= ϵ] >= 1- δ

- Probably Approximately Correct (PAC) learning defines a mathematical relationship between the **number of training samples, the error rate, and the probability that the available training data are large enough to attain the desired error rate**.
- PAC theory is concerned with the confidence with which we can say that our estimate is correct.

Hypothesis Space

What is hypothesis?

- **Machine Learning** helps us to predict results based on past experiences.
- ML professionals make an initial assumption for the solution of the problem.
- This assumption in Machine learning is known as Hypothesis.
- A Hypothesis is an assumption made by scientists, whereas a model is a mathematical representation that is used to test the hypothesis.

Contd..

- The hypothesis is defined as the proposed explanation based on insufficient evidence or assumptions.
- It is just a guess based on some known facts but has not yet been proven.
- **Example**: Let's understand the hypothesis with a common example. Some scientist claims that ultraviolet (UV) light can damage the eyes then it may also cause blindness.
- In this example, a scientist just claims that UV rays are harmful to the eyes, but we assume they may cause blindness.
- However, it may or may not be possible. Hence, these types of assumptions are called a hypothesis.

- A hypothesis in machine learning is the model's presumption regarding the connection between the input features and the result.

- Hypothesis space is the set of all the possible legal hypothesis.
- This is the set from which the machine learning algorithm would determine the best possible (only one) which would best describe the target function or the outputs.

Inductive Bias

Inductive Learning

- Inductive Machine Learning is a type of Machine Learning in which a model is trained on a dataset to make predictions based on previously unseen data.
- Inductive learning is used in a variety of applications namely, Natural Language Processing, and Computer Vision.
- Inductice Bias is mostly used for generalizing & represented by letter ‘L’.

Inductive Bias

- Need to make assumptions
  - Experience alone doesn't allow us to make conclusions about unseen data instances
- Two types of bias:
  - Restriction: Limit the hypothesis space
  - Preference: Impose ordering on hypothesis space

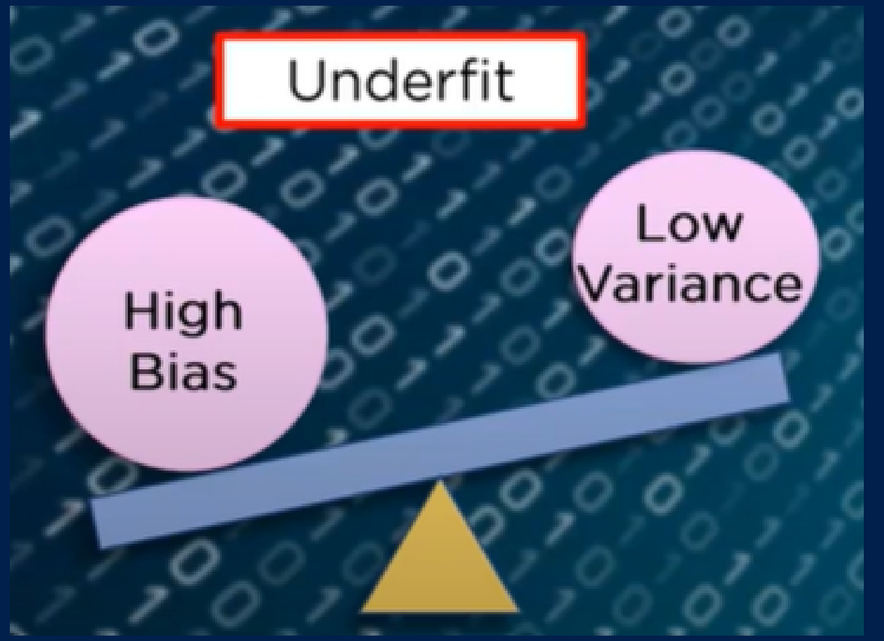
Generalization

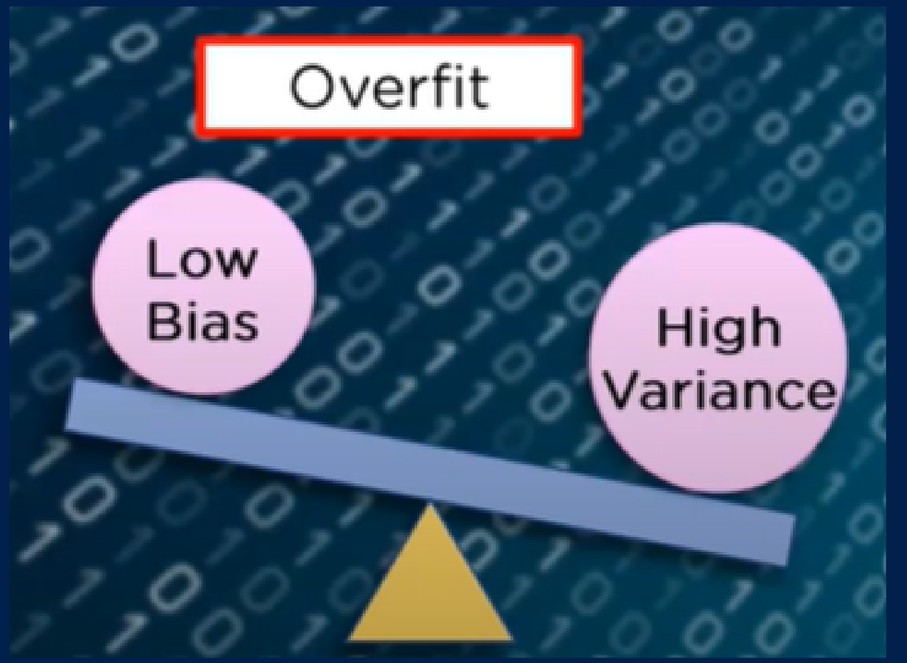
Generalization

- An ideal or good machine learning model should be able to perform well with new input data, allowing us to make accurate predictions about future data that the model has not seen before.
- This ability to work well with future data (unseen data) is known as generalization.

- If you train a model too well on training data, it will be incapable of generalizing.
- In such cases, it will end up making erroneous predictions when it’s given new data.
- This would make the model ineffective even though it’s capable of making correct predictions for the training data set.
- This is known as overfitting.
- New data may not have the exact same features and the model won’t be able to predict on it very well.

- **The inverse (underfitting) is also true, which happens when you train a model with inadequate data.**
- **In cases of underfitting, your model would fail to make accurate predictions even with the training data.**
- **This would make the model just as** useless **as overfitting.**
- **You would ideally want to choose a model that stands at the spot between overfitting and underfitting.**

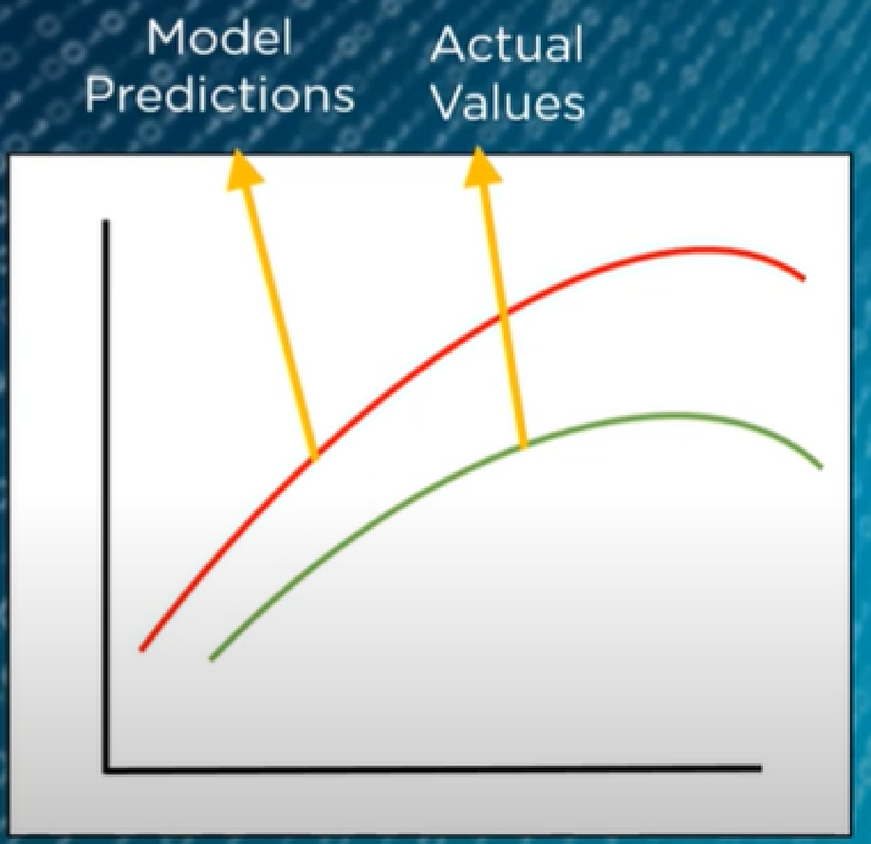




Bias Variance trade off

Importance of Error Calculation

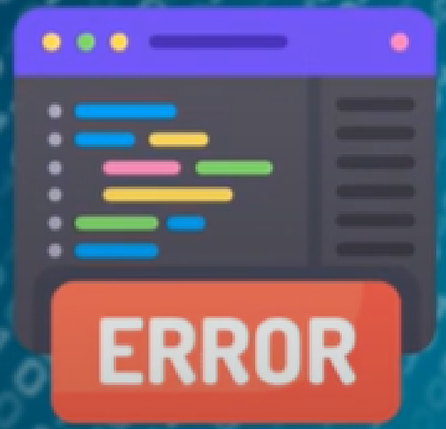
In Machine Learning, error is used to see how accurately our model can predict on data it uses to learn; as well as new, unseen data



Errors in Machine Learning

There are two main types of errors present in and machine learning model.
They are Reducible Errors and Irreducible Errors

Irreducible errors are errors which will always be present in a machine learning model, due to unknown variables, and who's values cannot be reduced



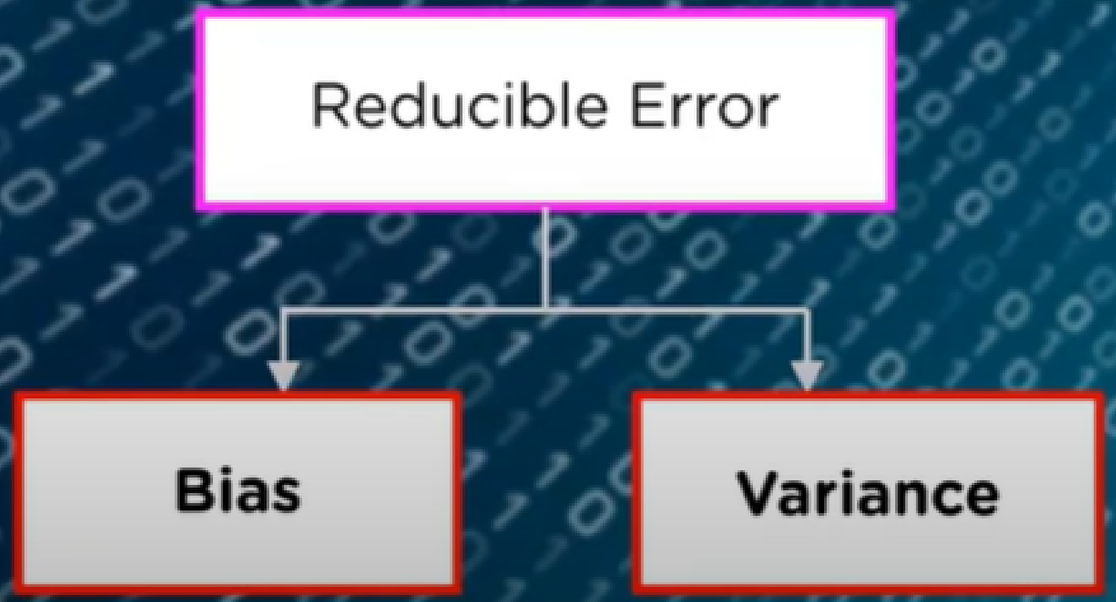
Errors in Machine Learning

There are two main types of errors present in and machine learning model.
They are Reducible Errors and Irreducible Errors

Reducible errors are those errors whose values can be further reduced to improve a model. They are caused because our model's output function does not match the desired output function and can be optimized

Errors in Machine Learning

Reducible Errors can be further divided into its two main constituent errors



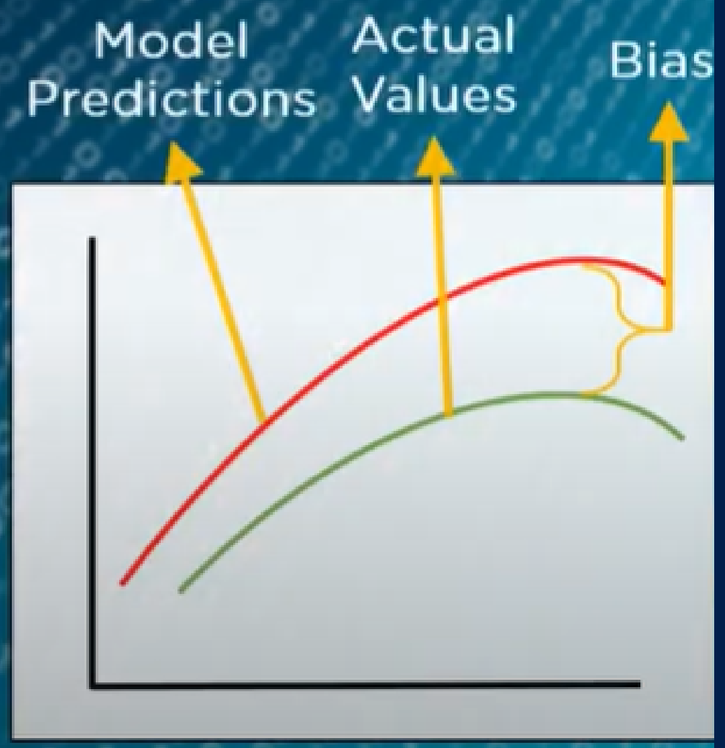
Bias and its effects

Bias is the difference between our actual and predicted values. Bias are the simple assumptions that our model makes about our data to be able to predict on new data

Below example shows our model making wrong assumptions and mistaking a Cat for a Fox

Actual

Predicted

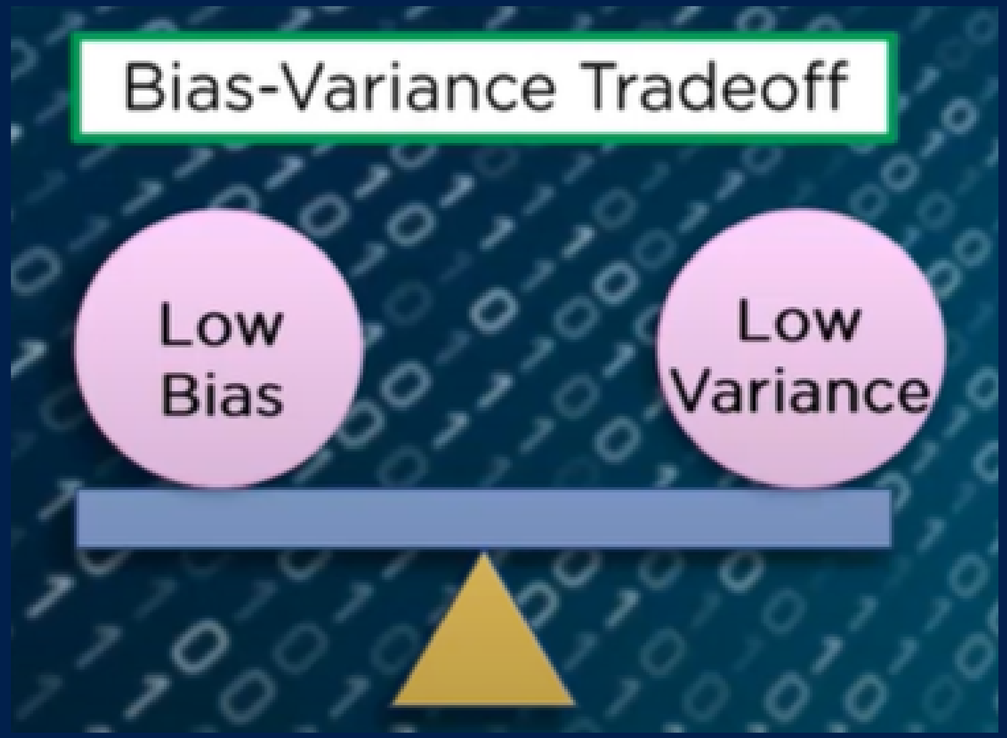


Bias & Variance

- Bias - Training data error
- Variance - Testing data error
- Variance can be defined as the model’s sensitivity to fluctuations in the data. Our model may learn from noise. This will cause our model to consider trivial features as important.
- When the bias is high, assumptions made by our model are too basic, and the model can’t capture the important features of the data.
- When the variance is high, our model model will capture all the features of the data given to it, will tune itself to the data, & predict on it very well.

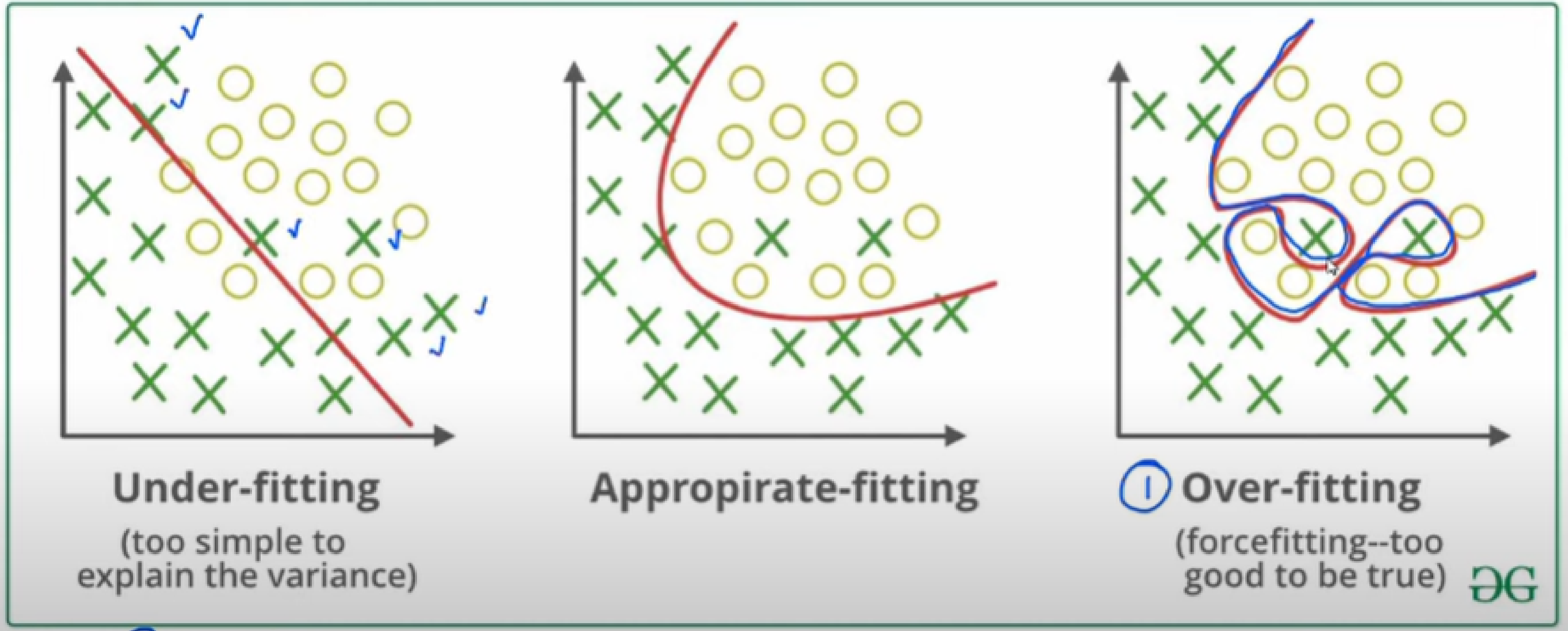
Bias & Variance Trade off

- To optimize the error in our model, we need to find the right balance between bias and variance.
- This is called bias variance trade off (Balanced model).



Bias vs Variance

Data { Training ⇒ excellent ⇒ Low bias
Test



2 training ⇒ poor