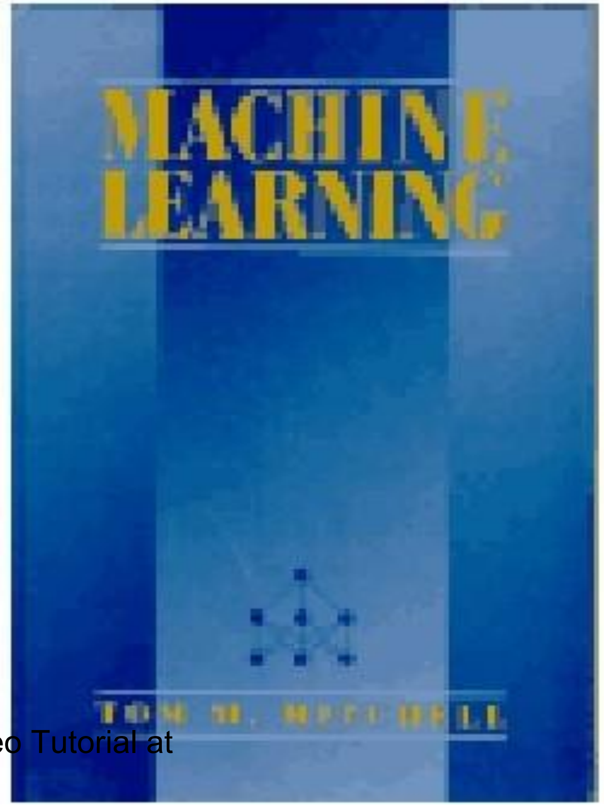
Introduction to Machine Learning

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Books

Machine Learning by Tom M. Mitchell



Syllabus

Module – 1 **10 Hours**

What is artificial intelligence? Problems, problem spaces and search, Heuristic search techniques. Textbook 1: Chapter 1, 2 and 3 RBT: L1, L2

Module – 2 **10 Hours**

Knowledge representation issues, Predicate logic, Representation knowledge using rules. Concept Learning: Concept learning task, Concept learning as search, Find-S algorithm, Candidate Elimination Algorithm, Inductive bias of Candidate Elimination Algorithm. Textbook 1: Chapter 4, 5 and 6 Texbook2: Chapter 2 (2.1-2.5, 2.7) RBT: L1, L2, L3

Module – 3 **08 Hours**

Decision Tree Learning: Introduction, Decision tree representation, appropriate problems, ID3 algorithm. Artificial Neural Network: Introduction, NN representation, appropriate problems, Perceptions, Back propagation algorithm.

Texbook2: Chapter 3 (3.1-3.4), Chapter 4 (4.1-4.5) RBT: L1, L2, L3

Module – 4 **10 Hours**

Bayesian Learning: Introduction, Bayes theorem, Bayes theorem and concept learning, ML and LS error hypothesis, ML for predicting, MDL principle, Bates optimal classifier, Gibbs algorithm, Naive Bayes classifier, BBN, EM Algorithm

Texbook2: Chapter 6 RBT: L1, L2, L3

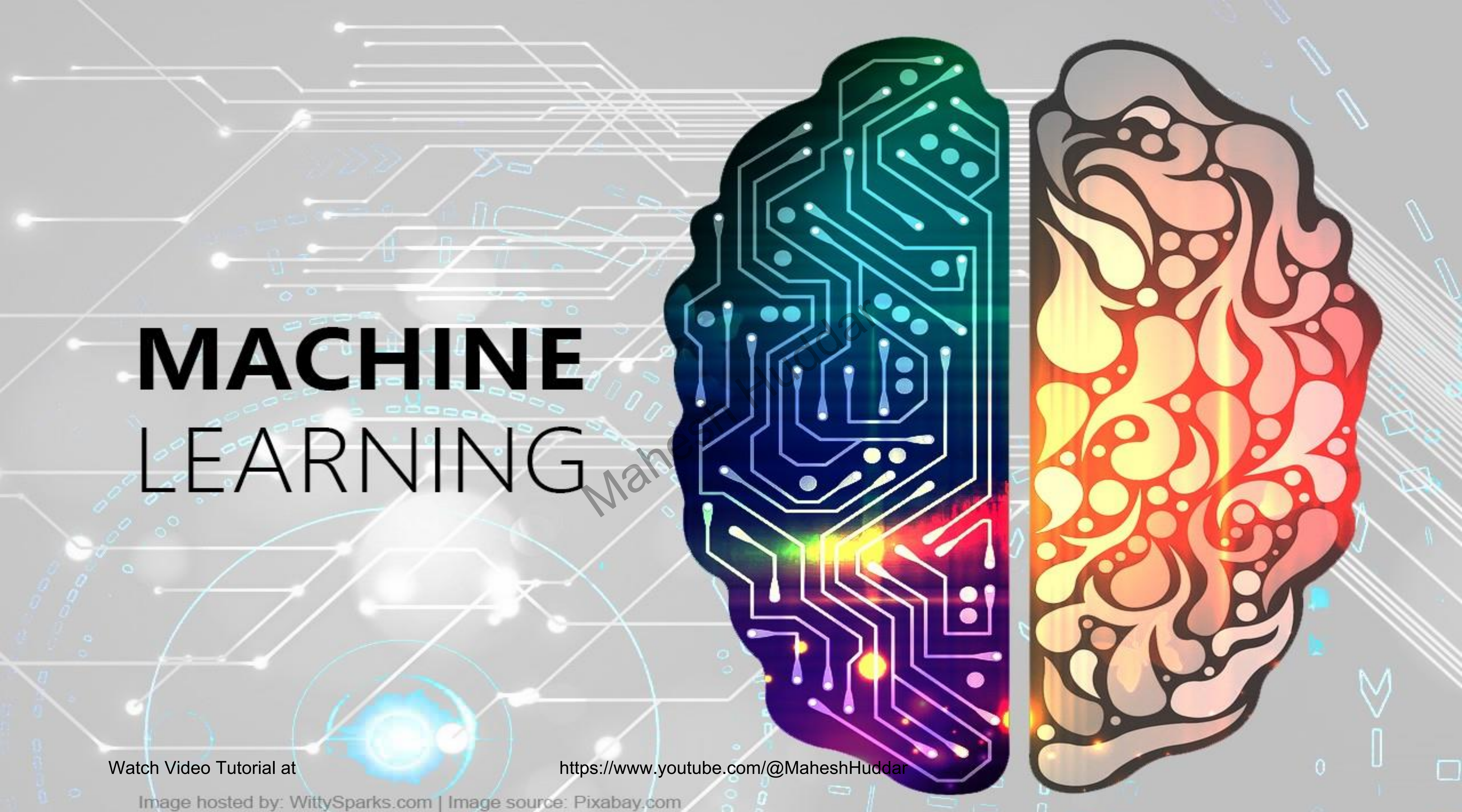
Module – 5 **12 Hours**

Instance-Base Learning: Introduction, k-Nearest Neighbor Learning, Locally weighted regression, Radial basis function, Case-Based reasoning. Reinforcement Learning: Introduction, The learning task, Q-Learning.

Textbook 1: Chapter 8 (8.1-8.5), Chapter 13 (13.1 – 13.3) RBT: L1, L2, L3

Syllabus - Lab

1.	Implement A* Search algorithm.
2.	Implement AO* Search algorithm.
3.	For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
4.	Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
5.	Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
6.	Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
7.	Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
8.	Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
9.	Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs


MACHINE
LEARNING
Image hosted by: WittySparks.com | Image source: Pixabay.com

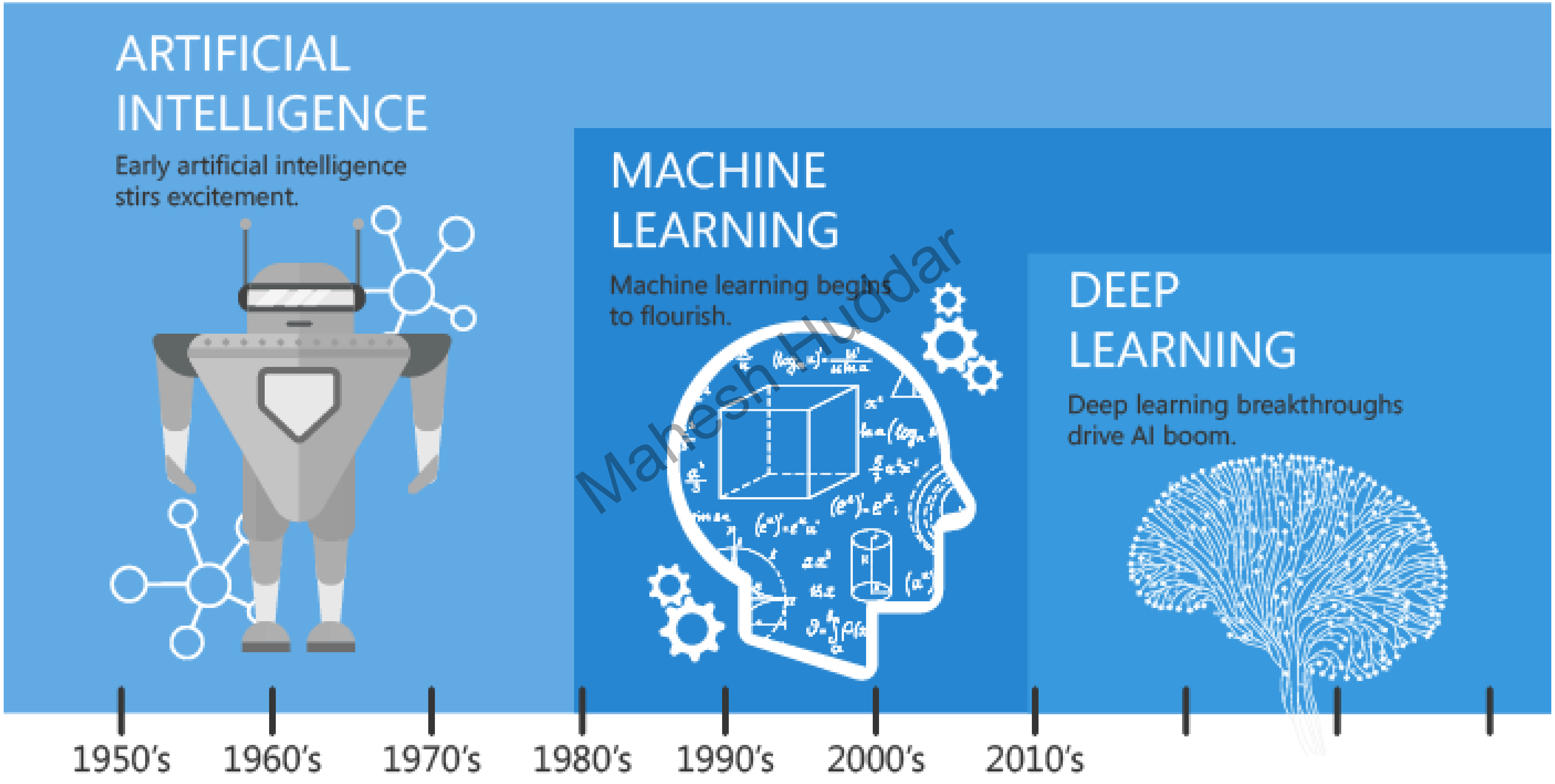
A Few Quotes

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing” (John Hennessy, President, Stanford)
- “Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Dir. Research, Yahoo)
- “Machine learning is going to result in a real revolution” (Greg Papadopoulos, CTO, Sun)
- “Machine learning is today’s discontinuity” (Jerry Yang, CEO, Yahoo)

Introduction to Machine Learning

- We have seen Machine Learning as a buzzword for the past few years, the reason for this might be the high amount of data production by applications, the increase of computation power in the past few years and the development of better algorithms.
- You may already be using a device or application that utilizes it.
- For example, GMAIL, WhatsApp, E-Commerce Website, Video Sharing Platforms, a wearable fitness tracker like Fitbit, or an intelligent home assistant like Google Home.

History of Machine Learning



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

What is Machine Learning..?

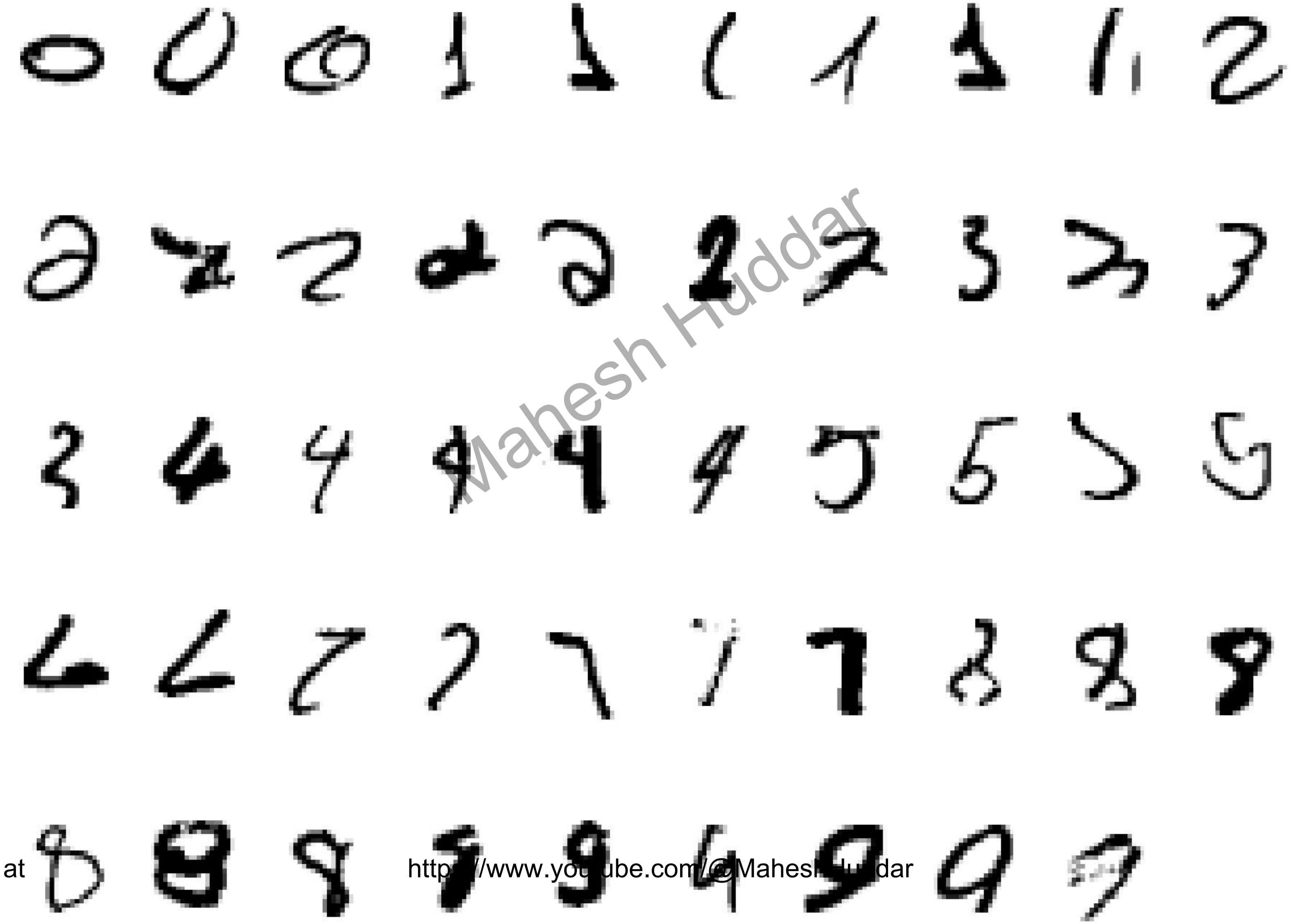
- “Learning is any process by which a system improves performance from experience.” - Herbert Simon

- A branch of **artificial intelligence**, concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data.

What is Machine Learning..?

- Definition by Tom Mitchell (1998):
- Machine Learning is the study of algorithms that
  - improve their performance P
  - at some task T
  - with experience E.
- A well-defined Machine Learning task is given by <P, T, E>.

A classic example of a task that requires machine learning



Handwritten Digit Recognition Problem

- **Task T:** Recognizing and Classifying handwritten words within images
- **Performance P:** percent of words correctly classified
- **Experience E:** a database of handwritten words with given classifications

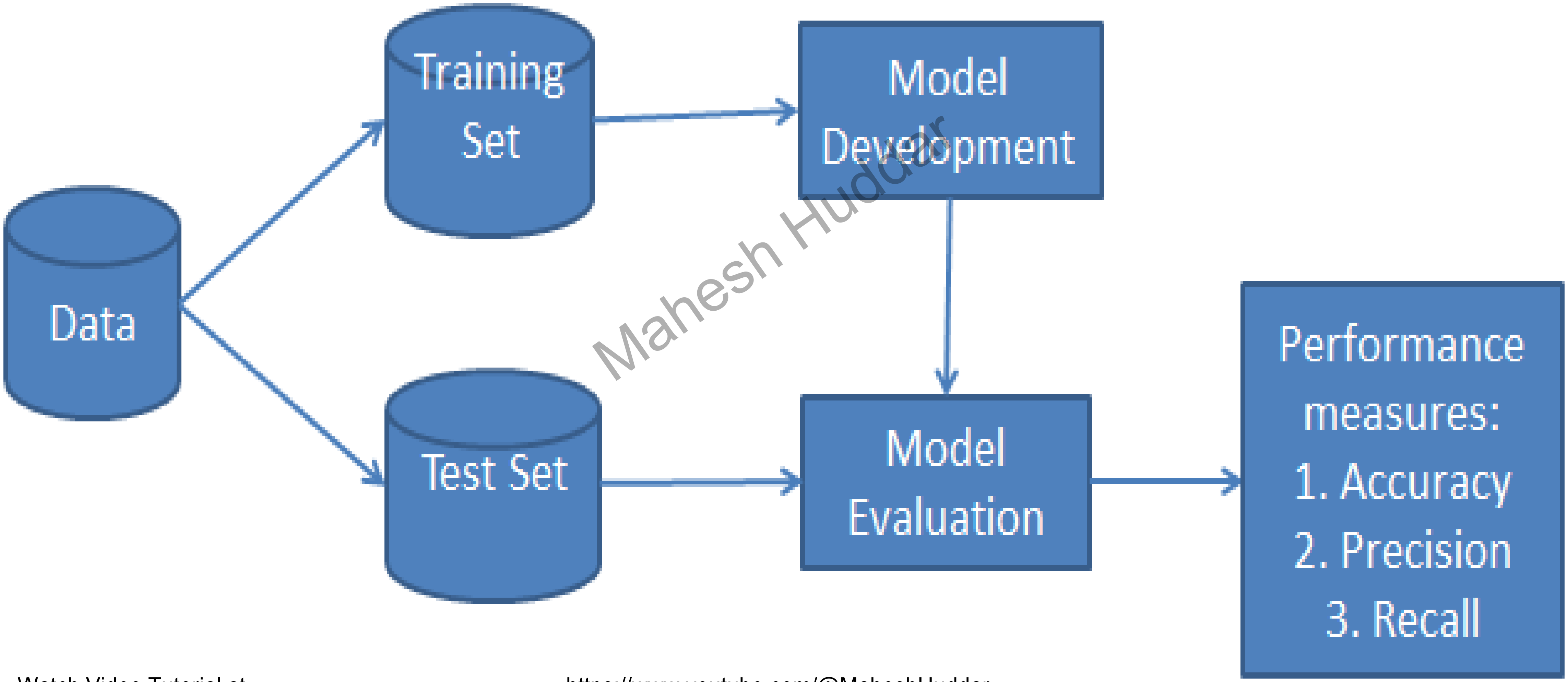
A robot driving learning problem

- **Task T:** driving on public four-lane highways using vision sensors
- **Performance P:** average distance traveled before an error (as judged by human overseer)
- **Experience E:** a sequence of images and steering commands recorded while observing a human driver

A checkers learning problem

- **Task T:** playing checkers
- **Performance P:** percent of games won against opponents
- **Experience E:** playing practice games against itself

How Machine Learning Works?

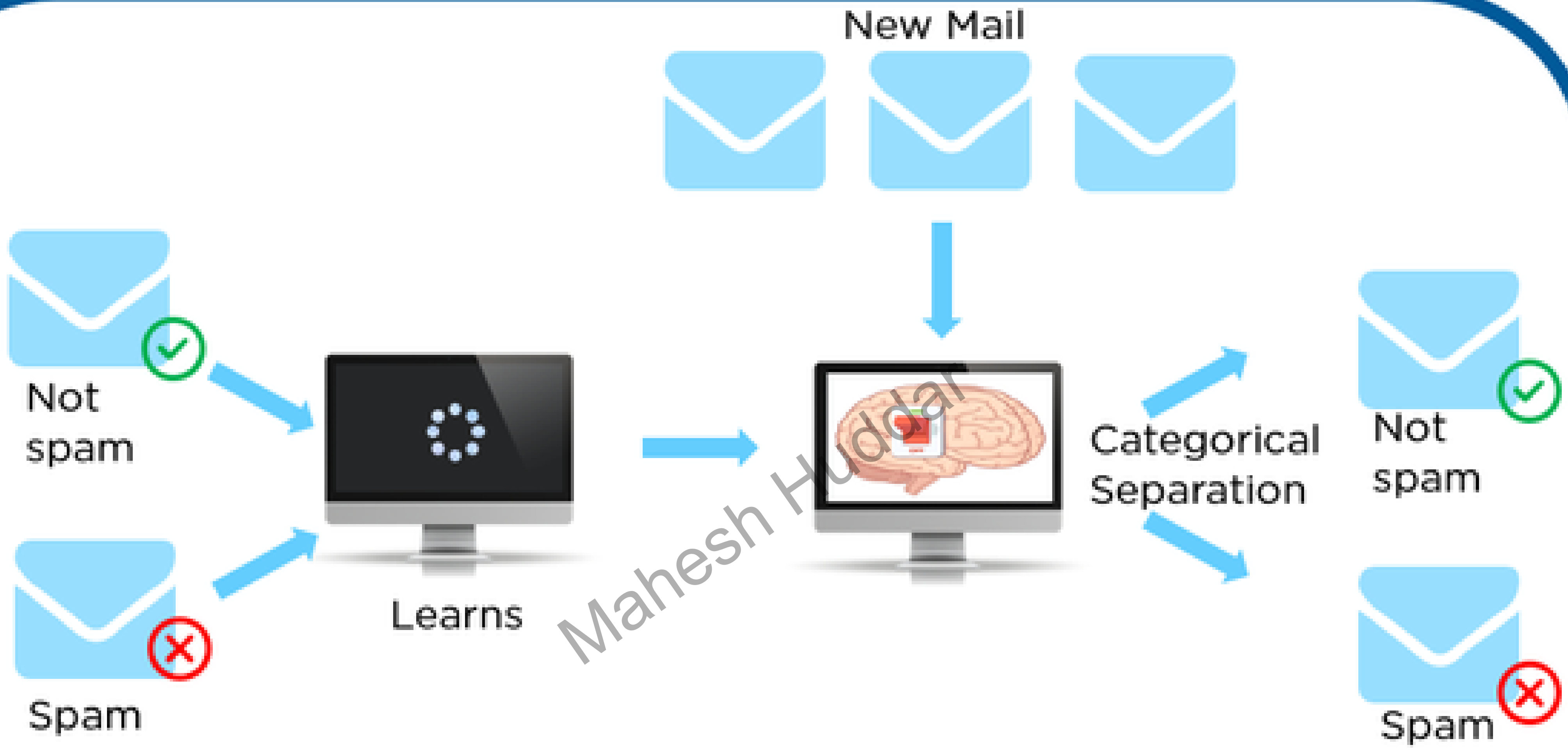


Types of Learning Algorithms

- **Supervised learning**
- **Unsupervised learning**
- **Reinforcement learning**

Supervised Learning

- In Supervised learning, an AI system is presented with data which is labeled, which means that each data tagged with the correct label.
- The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.
- As shown in the example, we have initially taken some data and marked them as 'Spam' or 'Not Spam'. This labeled data is used by the training supervised model, this data is used to train the model.
- Once it is trained we can test our model by testing it with some test new mails and checking of the model is able to predict the right output.

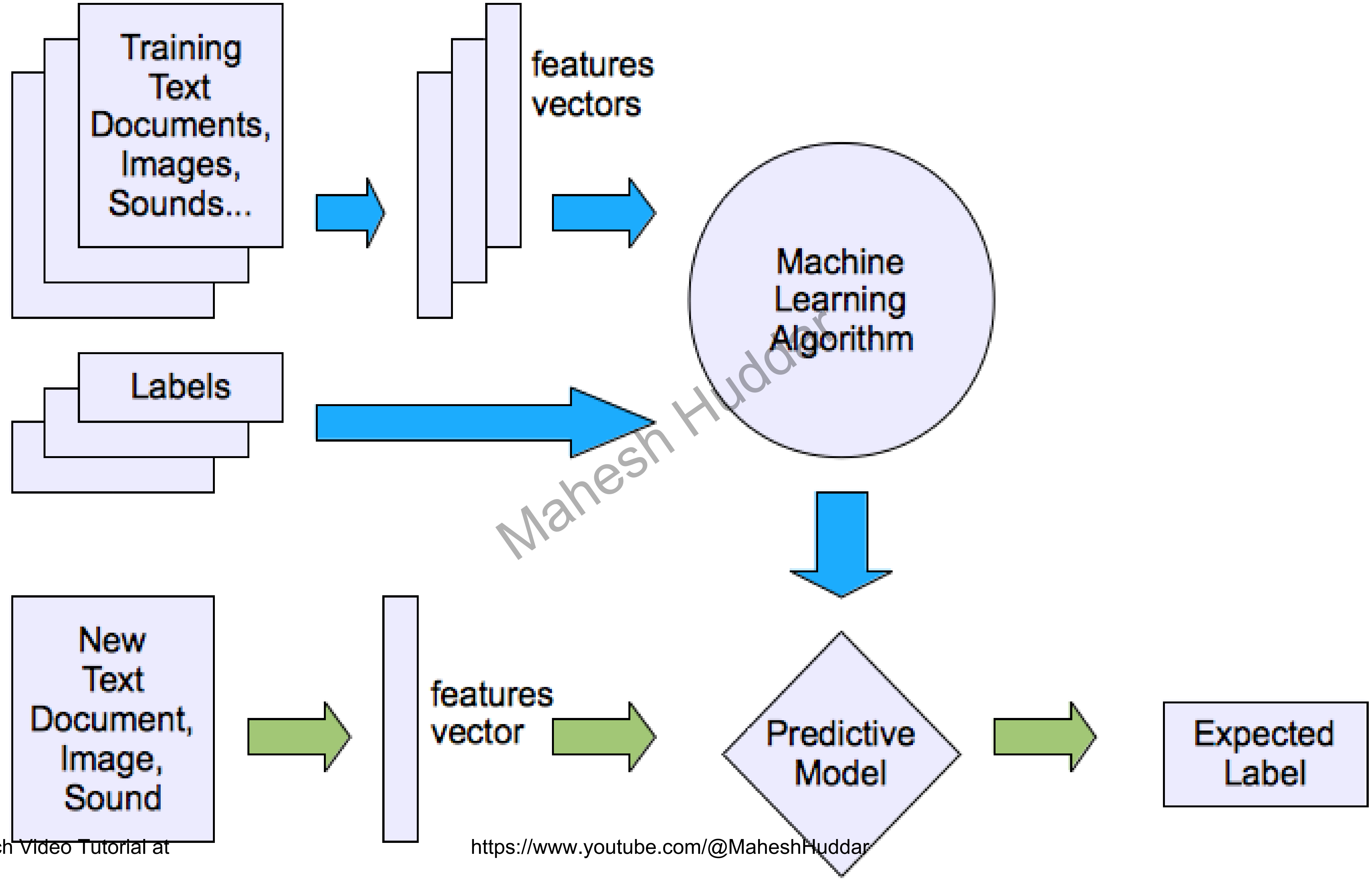


Enables the machine to be trained to classify observations into some class

Types of Supervised learning

- **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.

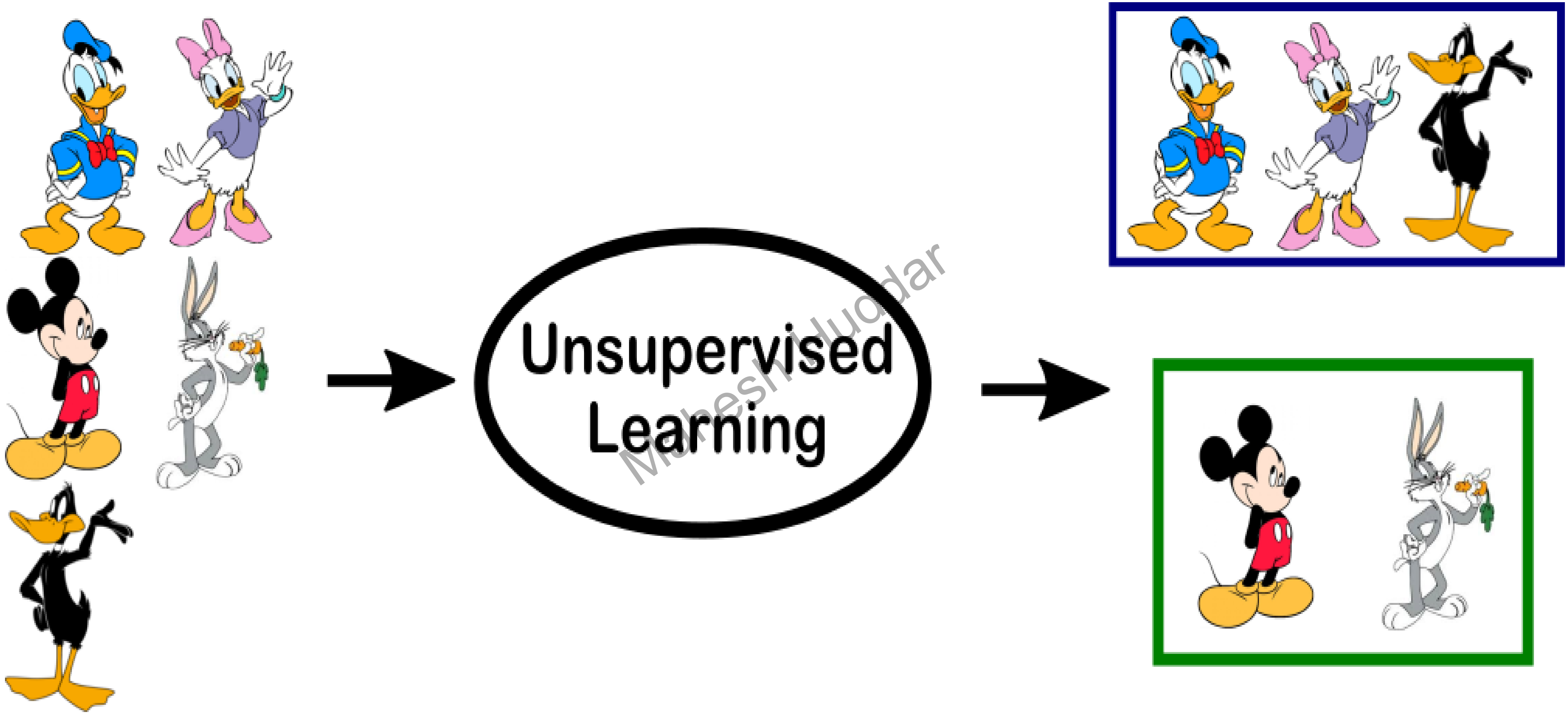
- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.


Training Text Documents, Images, Sounds...
features vectors
Machine Learning Algorithm
Labels
New Text Document, Image, Sound
features vector
Predictive Model
Expected Label

Supervised Learning

Unsupervised Learning Algorithm

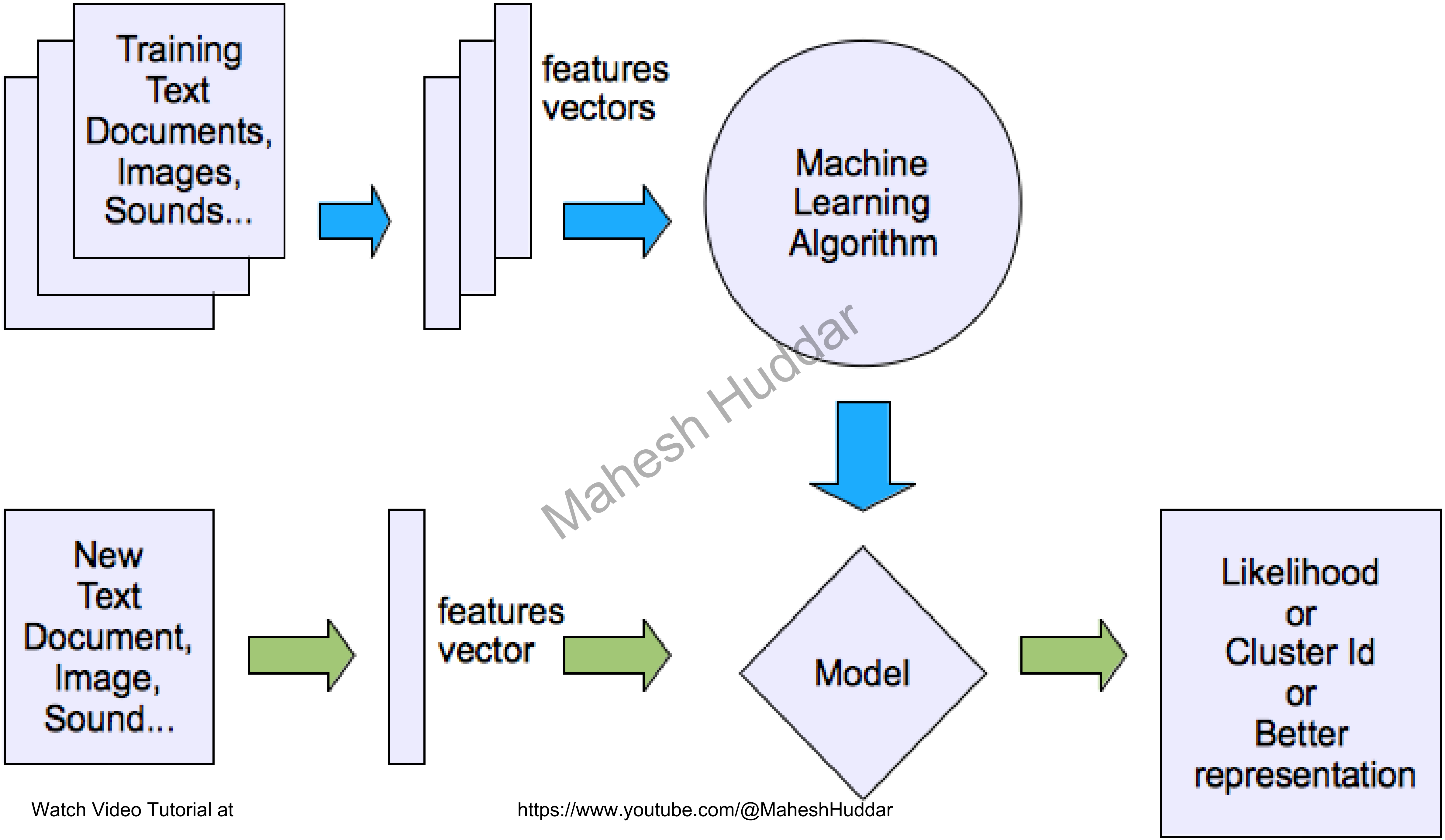
- In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system's algorithms act on the data without prior training.
- In the example, we have given some characters to our model which are 'Ducks' and 'Not Ducks'.
- In our training data, we don't provide any label to the corresponding data.
- The unsupervised model is able to separate both the characters by looking at the type of data and models the underlying structure or distribution in the data in order to learn more about it.


Unsupervised
Learning

Types of Unsupervised learning

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.


Training Text Documents, Images, Sounds...
features vectors
Machine Learning Algorithm
New Text Document, Image, Sound...
features vector
Model
Likelihood or Cluster Id or Better representation

Unsupervised Learning

Decision Function / Hypothesis

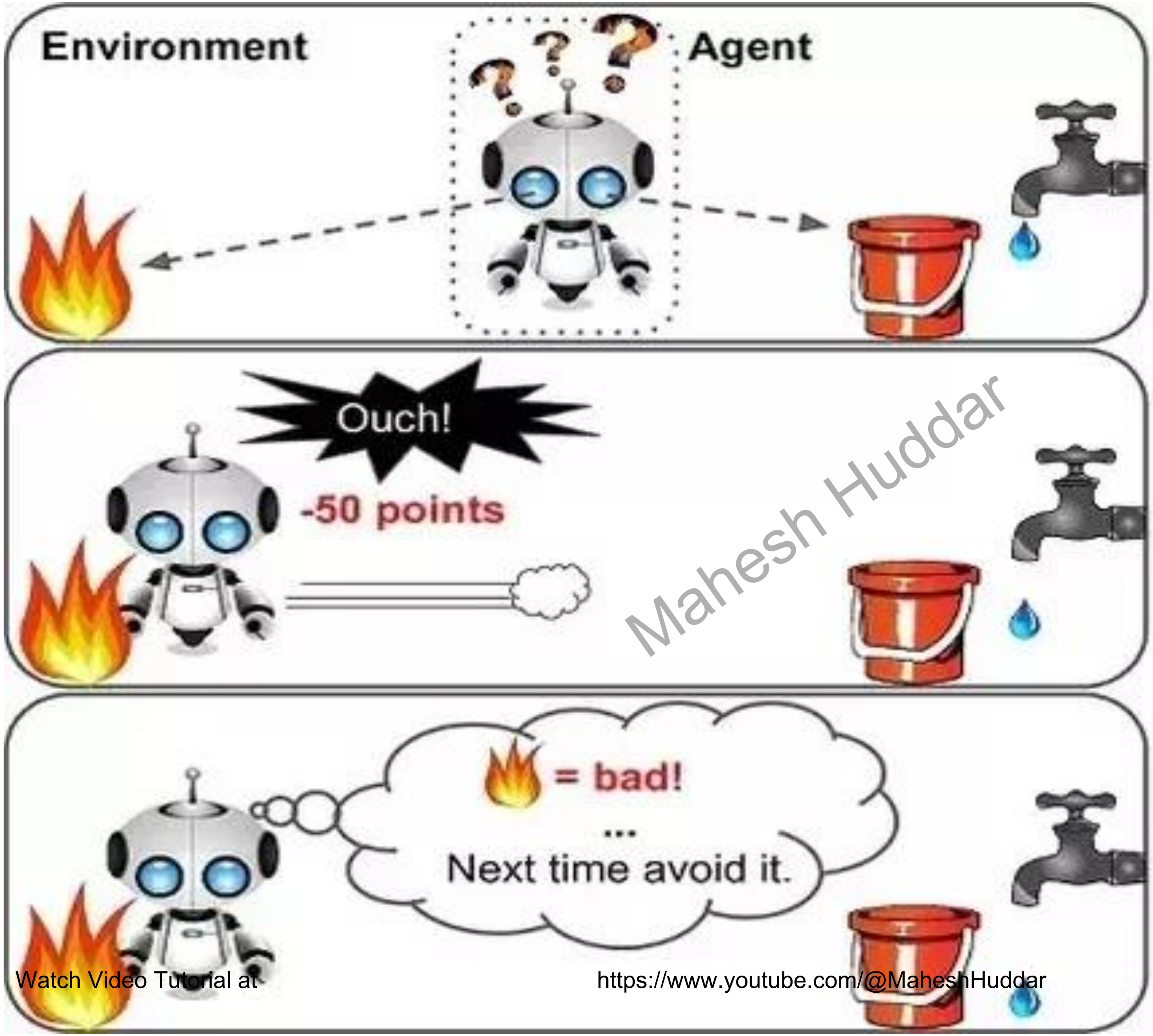
Unsupervised Classification

Reinforcement Learning

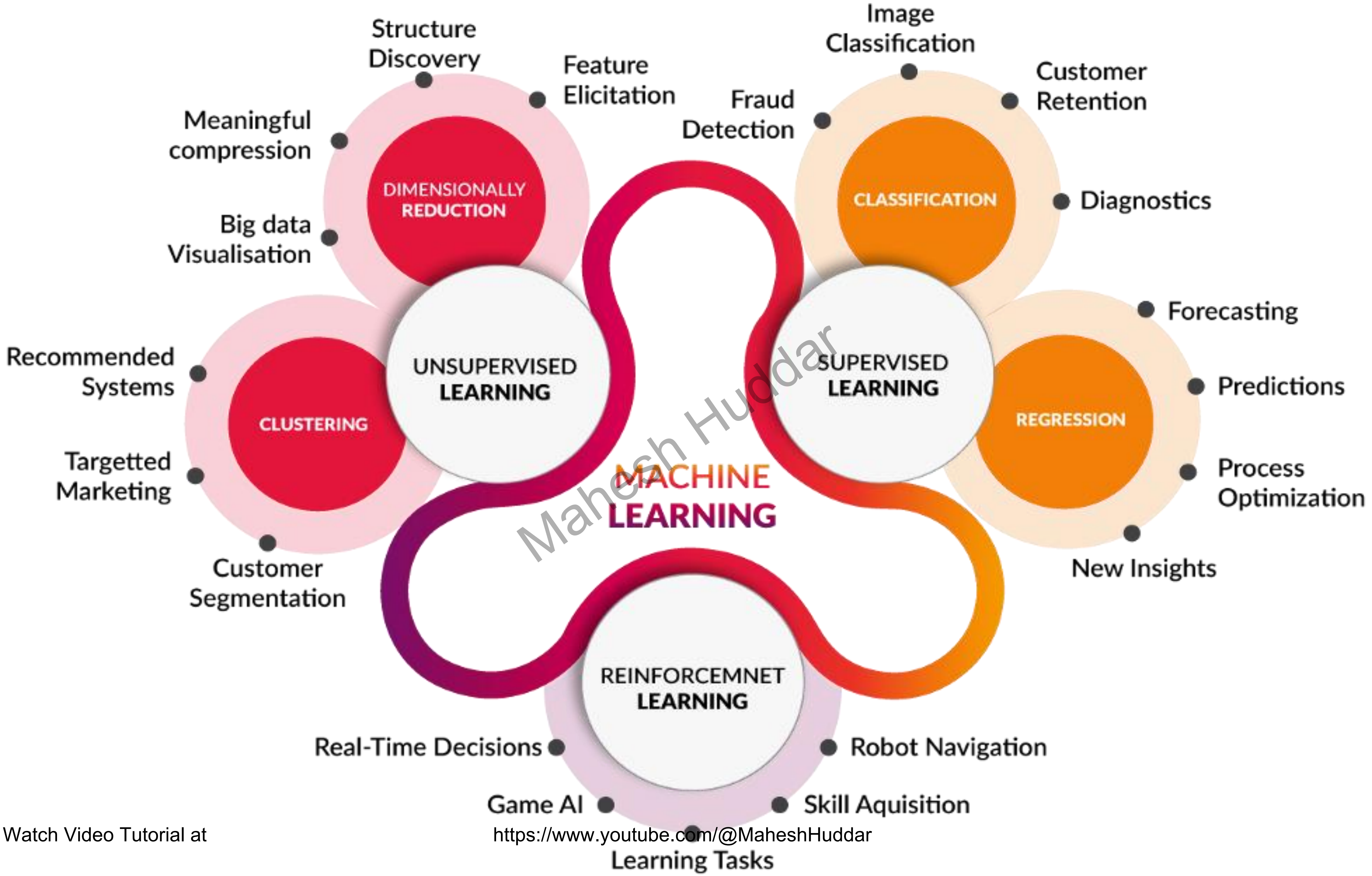
- A reinforcement learning algorithm, or agent, learns by interacting with its environment.
- The agent receives rewards by performing correctly and penalties for performing incorrectly.
- The agent learns without intervention from a human by maximizing its reward and minimizing its penalty.
- It is a type of dynamic programming that trains algorithms using a system of reward and punishment.

Reinforcement Learning

- In the example, we can see that the agent is given 2 options i.e. a path with water or a path with fire.
- A reinforcement algorithm works on reward a system i.e. if the agent uses the fire path then the rewards are subtracted and agent tries to learn that it should avoid the fire path.
- If it had chosen the water path or the safe path then some points would have been added to the reward points, the agent then would try to learn what path is safe and what path isn't.
- It is basically leveraging the rewards obtained, the agent improves its environment knowledge to select the next action.



1. Observe
2. Select action using policy
3. Action!
4. Get reward or penalty
5. Update policy (learning step)
6. Iterate until an optimal policy is found


Structure Discovery
Feature Elicitation
Meaningful compression
DIMENSIONALLY REDUCTION
Big data Visualisation
Recommended Systems
CLUSTERING
Targetted Marketing
Customer Segmentation
UNSUPERVISED LEARNING
Image Classification
Customer Retention
Fraud Detection
CLASSIFICATION
Diagnostics
SUPERVISED LEARNING
Forecasting
Predictions
REGRESSION
Process Optimization
New Insights
MACHINE LEARNING
REINFORCEMNET LEARNING
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

Applications of Machine Learning

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual sequences of credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant or unusual sound in your car engine.
- Prediction:
  - Future stock prices or currency exchange rates

Influence of Disciplines on Machine Learning

Artificial intelligence

- Learning symbolic representations of concepts. Machine learning as a search problem. Learning as an approach to improving problem solving. Using prior knowledge together with training data to guide learning.

Bayesian methods

- Bayes' theorem as the basis for calculating probabilities of hypotheses. The naive Bayes classifier. Algorithms for estimating values of unobserved variables.

Computational complexity theory

- Theoretical bounds on the inherent complexity of different learning tasks, measured in terms of the computational effort, number of training examples, number of mistakes, etc. required in order to learn.

Control theory

- Procedures that learn to control processes in order to optimize predefined objectives and that learn to predict the next state of the process they are controlling.

Influence of Disciplines on Machine Learning

Information theory

- Measures of entropy and information content. Minimum description length approaches to learning. Optimal codes and their relationship to optimal training sequences for encoding a hypothesis.

Philosophy

- Occam's razor, suggesting that the simplest hypothesis is the best. Analysis of the justification for generalizing beyond observed data.

Psychology and neurobiology

- The power law of practice, which states that over a very broad range of learning problems,
- people's response time improves with practice according to a power law. Neurobiological studies motivating artificial neural network models of learning.

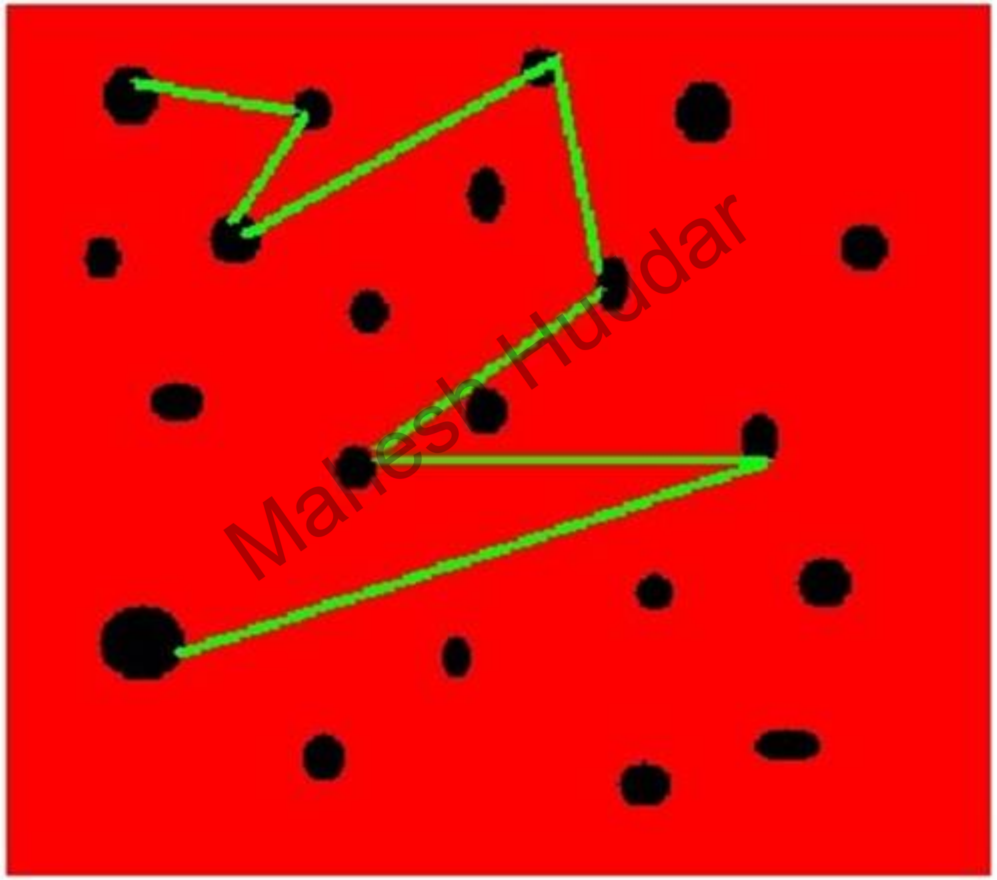
Statistics

- Characterization of errors (e.g., bias and variance) that occur when estimating the accuracy of a hypothesis based on a limited sample of data. Confidence intervals, statistical tests.

CONCEPT LEARNING

- The problem of inducing general functions from specific training examples is central to learning.
- Concept learning can be formulated as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples.
- What is Concept Learning…?
- **“A task of acquiring potential hypothesis (solution) that best fits the given training examples.”**

CONCEPT LEARNING



A CONCEPT LEARNING TASK

- Consider the example task of learning the target concept "**days on which XYZ enjoys his favorite water sport.**"
- Table describes a set of example days, each represented by a set of ***attributes.*** The attribute ***EnjoySport*** indicates whether or not XYZ enjoys his favorite water sport on this day.
- The task is to learn to predict the value of ***EnjoySport*** for an arbitrary day, based on the values of its other attributes.

Objective is to learn EnjoySport

{Sky, AirTemp, Humidity, Wind, Water, Forecast} → EnjoySport

Tom enjoys his favorite water sports

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

Objective is to learn EnjoySport

{Sky, AirTemp, Humidity, Wind, Water, Forecast} → EnjoySport

Input variables

Output

<x1, x2, x3, x4, x5, x6> → <y>

A CONCEPT LEARNING TASK - Notation

Instances (X)

Target Concept (C)

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

Training examples (D)

A CONCEPT LEARNING TASK

- What hypothesis representation shall we provide to the learner in this case?
- Let us begin by considering a simple representation in which each hypothesis consists of a conjunction of constraints on the instance attributes.
- In particular, let each hypothesis be a vector of six constraints, specifying the values of the six attributes **Sky, AirTemp, Humidity, Wind, Water,** and **Forecast.**
- For each attribute, the hypothesis will either
  - indicate by a **"?'** that any value is acceptable for this attribute,
  - specify a single required value (e.g., **Warm)** for the attribute, or
  - indicate by a **"ø"** that no value is acceptable.

A CONCEPT LEARNING TASK

- If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example (h(x) = 1).
- To illustrate, the hypothesis that Prabhas enjoys his favorite sport only on cold days with high humidity (independent of the values of the other attributes) is represented by the expression

A CONCEPT LEARNING TASK

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

1. **(Sunny, Warm, Normal, Strong, Warm, Same)**
2. **(Rainy, Warm, High, Strong, Warm, Same)**
3. **(Rainy, Cold, High, Strong, Warm, Change)**
4. **(?, Cold, High, ?, ?, ?)**

A CONCEPT LEARNING TASK

- The most general hypothesis-that every day is a positive example-is represented by

(?, ?, ?, ?, ?, ?)

- and the most specific possible hypothesis-that no day is a positive example-is represented by

(ø, ø, ø, ø, ø, ø)

A CONCEPT LEARNING TASK - Notation

Instances (X)

Target Concept (C)

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

Training examples (D)

A CONCEPT LEARNING TASK - Notation

- **Given:**
  - Instances X: Possible days, each described by the attributes
    - *Sky* (with possible values *Sunny*, *Cloudy*, and *Rainy*),
    - *AirTemp* (with values *Warm* and *Cold*),
    - *Humidity* (with values *Normal* and *High*),
    - *Wind* (with values *Strong* and *Weak*),
    - *Water* (with values *Warm* and *Cool*), and
    - *Forecast* (with values *Same* and *Change*).
  - Hypotheses H: Each hypothesis is described by a conjunction of constraints on the attributes *Sky*, *AirTemp*, *Humidity*, *Wind*, *Water*, and *Forecast*. The constraints may be "?" (any value is acceptable), "Ø" (no value is acceptable), or a specific value.
  - Target concept c: $EnjoySport : X \rightarrow \{0, 1\}$
  - Training examples D: Positive and negative examples of the target function (see Table 2.1).
- **Determine:**
  - A hypothesis h in H such that $h(x) = c(x)$ for all x in X.

A CONCEPT LEARNING TASK - Search

- Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.
- The goal of this search is to find the hypothesis that best fits the training examples.
- It is important to note that by selecting a hypothesis representation, the designer of the learning algorithm implicitly defines the space of all hypotheses that the program can ever represent and therefore can ever learn.

A CONCEPT LEARNING TASK - Search

Instance Space:

- Consider, for example, the instances X and hypotheses H in the ***EnjoySport*** learning task.
- Given that the attribute ***Sky*** has three possible values, and that ***AirTemp, Humidity, Wind, Water,*** and ***Forecast*** each have two possible values, the instance space X contains exactly **3 . 2 . 2 . 2 . 2 . 2 = 96** distinct instances.

A CONCEPT LEARNING TASK - Search

F1 - > A, B

F2 - > X, Y

3 * 3 = 9 + 1 = 10

(?, **ø)**

Instance Space: (A, X), (A, Y), (B, X), (B, Y) – **4 Instances**

Hypothesis Space: (A, X), (A, Y), (A, **ø**), (A, ?), (B, X), (B, Y), (B, **ø**), (B, ?), (**ø**, X), (**ø**, Y), (**ø**, **ø**), (**ø**, ?), (**?**, X), (**?**, Y), (**?**, **ø**), (**?**, ?) - **16**

Hypothesis Space: (A, X), (A, Y), (A, ?), (B, X), (B, Y), (B, ?), (**?**, X), (**?**, Y (**?**, ?) (**ø**, **ø**)- **10**

A CONCEPT LEARNING TASK – Instance Space

Suppose the attribute ***Sky*** has three possible values, and that ***AirTemp, Humidity, Wind, Water,*** and ***Forecast*** each have two possible values

Instance Space

Possible distinct instances = 3 * 2 * 2 * 2 * 2 * 2 = 96

A CONCEPT LEARNING TASK - Search

Hypothesis Space

- Similarly there are **5 . 4 . 4 . 4 . 4 . 4 = 5120** syntactically distinct hypotheses within H.
- Notice, however, that every hypothesis containing one or more "**ø**" symbols represents the empty set of instances; that is, it classifies every instance as negative.
- Therefore, the number of semantically distinct hypotheses is only **1 + (4 . 3 . 3 . 3 . 3 . 3) = 973.**
- Our EnjoySport example is a very simple learning task, with a relatively small, finite hypothesis space.

A CONCEPT LEARNING TASK – Hypothesis Space

Hypothesis Space: A set of all possible hypotheses

Possible syntactically distinct Hypotheses for *EnjoySport*

$= 5 * 4 * 4 * 4 * 4 * 4 = 5120$

- Sky has **three** possible values
- **Fourth** value don't care (?)
- **Fifth** value is empty set Ø

General-to-Specific Ordering of Hypotheses

- To illustrate the general-to-specific ordering, consider the two hypotheses

  h1 = (Sunny, ?, ?, Strong, ?, ?)

  h2 = (Sunny, ?, ?, ?, ?, ?)

- Now consider the sets of instances that are classified positive by hl and by h2. Because h2 imposes fewer constraints on the instance, it classifies more instances as positive.
- In fact, any instance classified positive by h1 will also be classified positive by h2. Therefore, we say that h2 is more general than h1.

More General Than hypothesis

- For any instance x in X and hypothesis h in H, we say that x satisjies h if and only if h(x) = 1.
- We define the ***more_general_than_or_equal**e_to* relation in terms of the sets of instances that satisfy the two hypotheses:

More General Than hypothesis

Given hypotheses **hj** and **hk**, **hj** is **more_general_than_or_equal_to hk** if and only if any instance that satisfies **hk** also satisfies ***hj.***

Definition: Let h_j and h_k be boolean-valued functions defined over X. Then h_j is **more_general_than_or_equal_to** h_k (written $h_j \geq_g h_k$) if and only if

$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

We can also say that ***hj*** is ***more_specific_than hk*** when ***hk*** is ***more_general_than hj.***

FIND-S: FINDING A MAXIMALLY SPECIFIC HYPOTHESIS

1. Initialize h to the most specific hypothesis in H
2. For each positive training instance x
   - For each attribute constraint a_i in h

     If the constraint a_i is satisfied by x

     Then do nothing

     Else replace a_i in h by the next more general constraint that is satisfied by x
3. Output hypothesis h

FIND-S Algorithm

Finding A Maximally Specific Hypothesis

Solved Example - 1

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

FIND-S: Step-1

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

1. Initialize *h* to the most specific hypothesis in *H*

h0 = **<Ø, Ø, Ø, Ø, Ø, Ø>**

FIND-S: Step-2

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

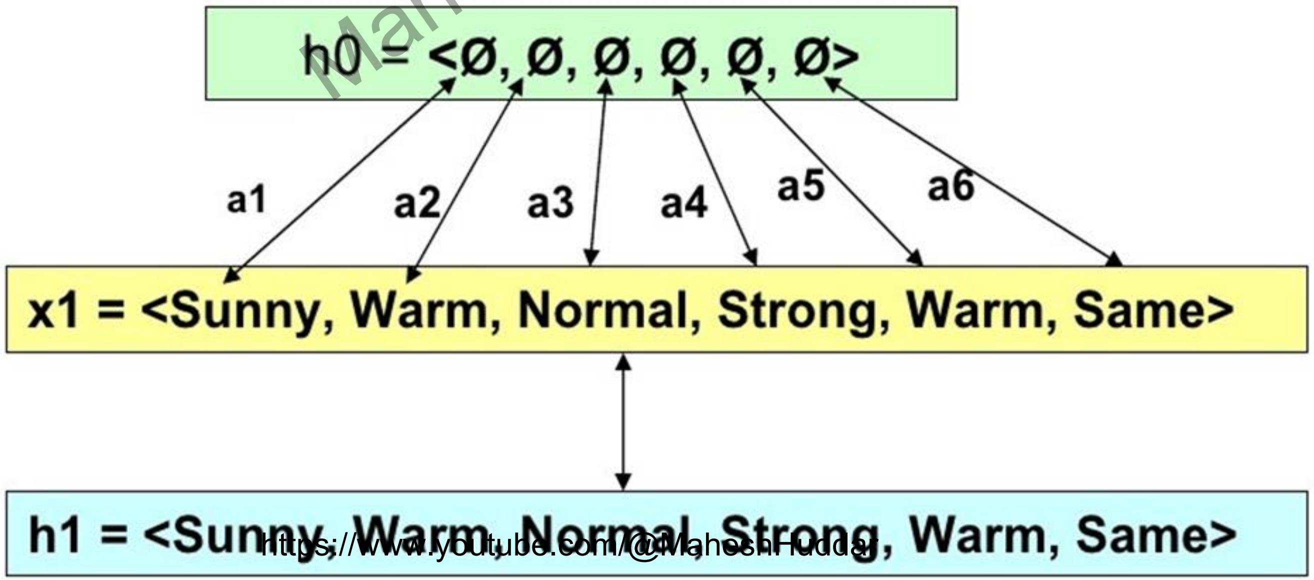
2. For each positive training instance x

- For each attribute constraint a_i in h

  If the constraint a_i is satisfied by x

  Then do nothing

  Else replace a_i in h by the next more general constraint that is satisfied by x



Iteration 1

FIND-S: Step-2

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

Iteration 2

h1 = <Sunny, Warm, Normal, Strong, Warm, Same>

x2 = <Sunny, Warm, High, Strong, Warm, Same>

h2 = <Sunny, Warm, ?, Strong, Warm, Same>

FIND-S: Step-2

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

Iteration 3

Ignore

h3 = <Sunny, Warm, ?, Strong, Warm, Same>

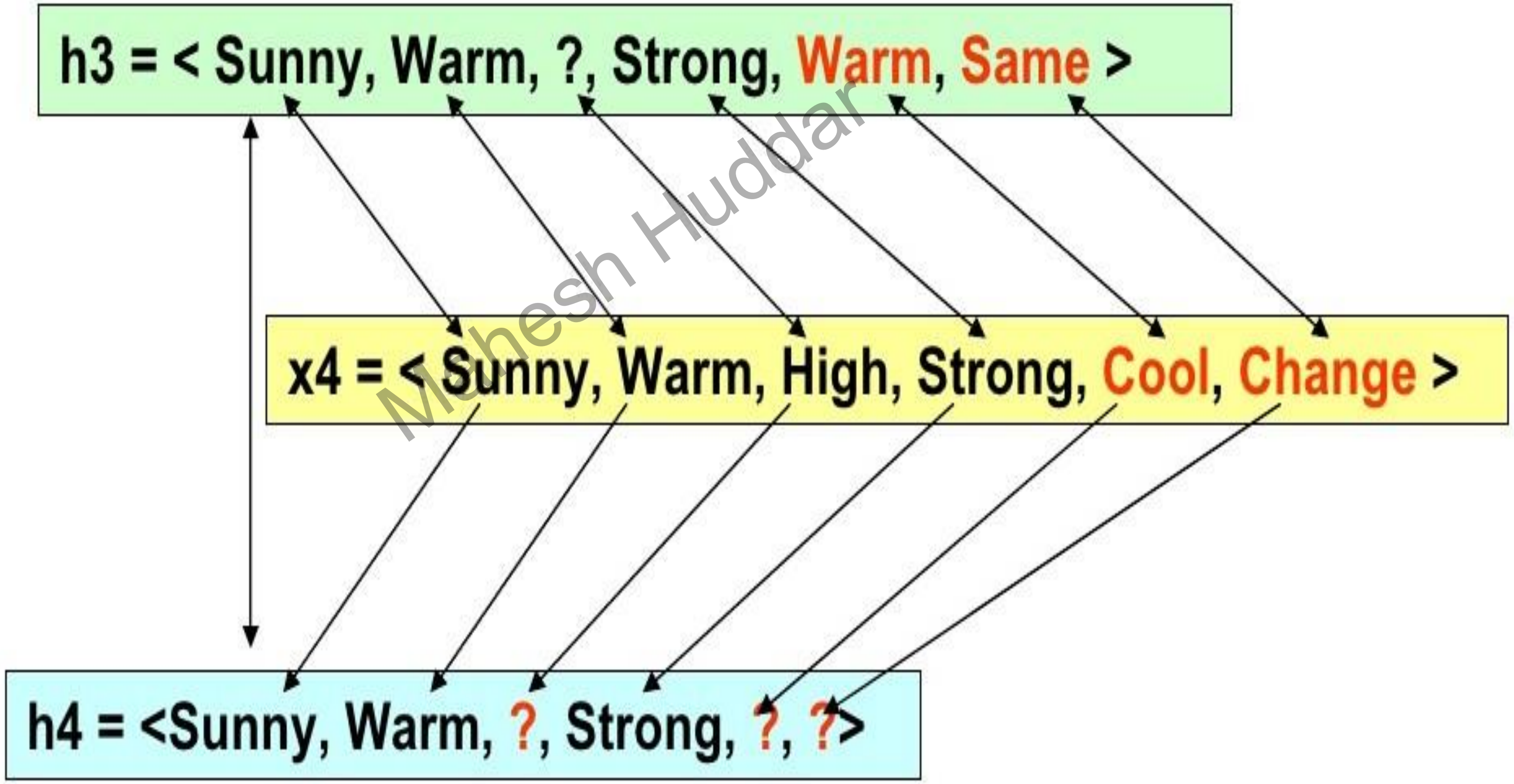
FIND-S: Step-2

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

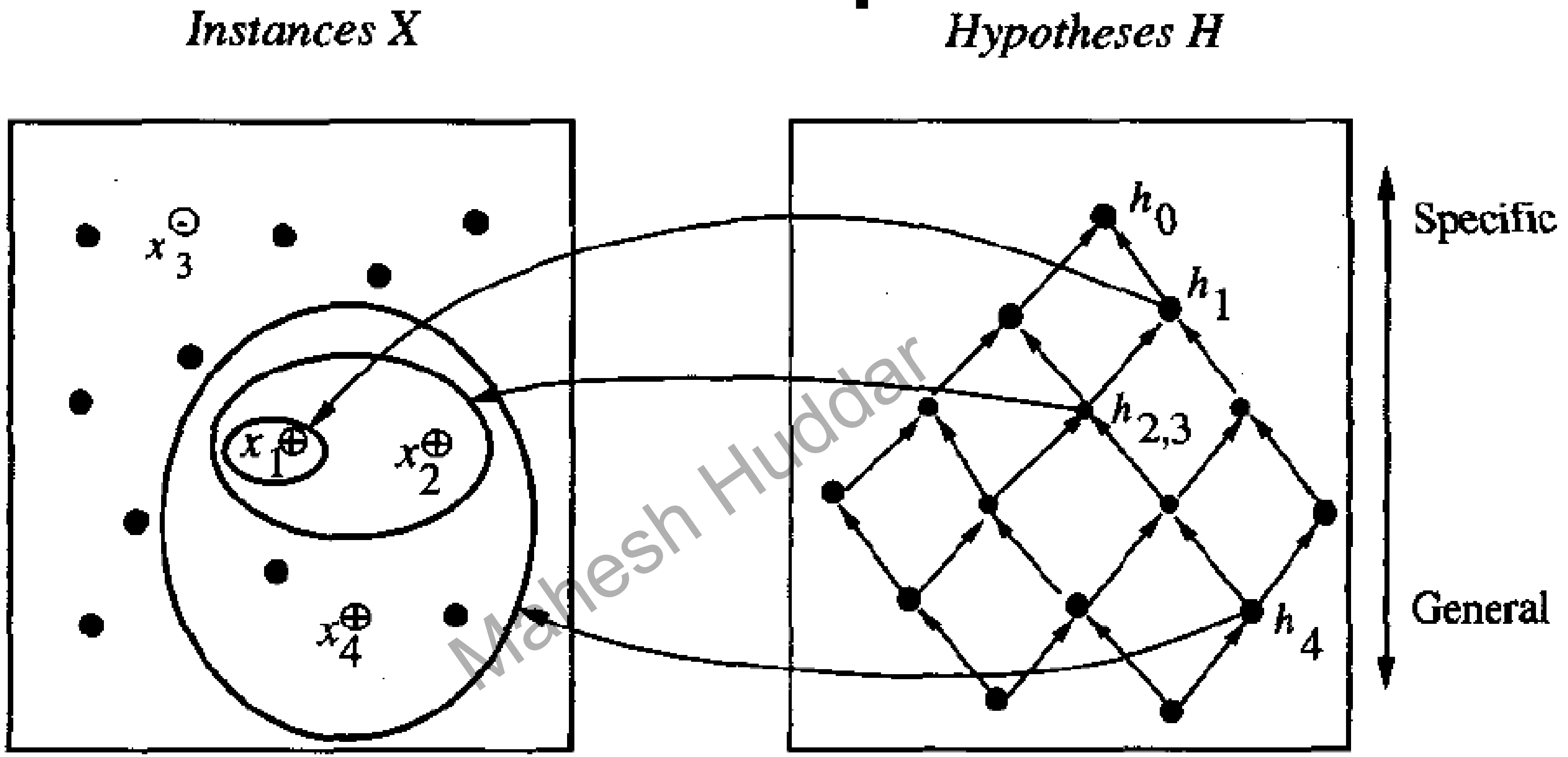
Iteration 4

Step 3

Output



FIND-S: Step-2



$h_0 = \langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle$

$x_1 = \langle$*Sunny Warm Normal Strong Warm Same*$\rangle$, +

$h_1 = \langle$*Sunny Warm Normal Strong Warm Same*$\rangle$

$x_2 = \langle$*Sunny Warm High Strong Warm Same*$\rangle$, +

$h_2 = \langle$*Sunny Warm ? Strong Warm Same*$\rangle$

$x_3 = \langle$*Rainy Cold High Strong Warm Change*$\rangle$, -

$h_3 = \langle$*Sunny Warm ? Strong Warm Same*$\rangle$

$x_4 = \langle$*Sunny Warm High Strong Cool Change*$\rangle$, +

$h_4 = \langle$*Sunny Warm ? Strong ? ?* $\rangle$

FIND-S Algorithm

Finding A Maximally Specific Hypothesis

Solved Example - 2

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

FIND-S Algorithm Solved Example - 2

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

1. **How many concepts are possible for this instance space?**
2. **How many hypotheses can be expressed by the hypothesis language?**
3. **Apply the FIND-S algorithm by hand on the given training set. Consider the examples in the specified order and write down your hypothesis each time after observing an example.**

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

1. How many concepts are possible for this instance space?

Solution: 2 * 3 * 2 * 2 * 3 = 72

2. How many hypotheses can be expressed by the hypothesis language?

Solution: 4 * 5 * 4 * 4 * 5 = 1600

Semantically Distinct Hypothesis = (3 * 4 * 3 * 3 * 4) + 1 = 433

Step 1:

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

h0 = (ø, ø, ø, ø, ø)

Step 2:

X1 = (some, small, no, expensive, many) – No

Negative Example Hence Ignore

h1 = (ø, ø, ø, ø, ø)

X2 = (many, big, no, expensive, one) – Yes

h2 = (many, big, no, expensive, one)

Step 2:

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

h2 = (many, big, no, expensive, one)

X3 = (some, big, always, expensive, few) – No

Negative example hence Ignore

h3 = (many, big, no, expensive, one)

X4 = (many, medium, no, expensive, many) – Yes

h4 = (many, ?, no, expensive, ?)

example	*citations*	*size*	*inLibrary*	*price*	*editions*	*buy*
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

Step 2:

h4 = (many, ?, no, expensive, ?)

X5 = (many, small, no, affordable, many) – Yes

h5 = (many, ?, no, ?, ?)

Step 3:

Final Hypothesis is:

h5 = (many, ?, no, ?, ?)

FIND-S Algorithm

Unanswered Questions

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Unanswered Questions of FIND-S Algorithm

1. Has the learner converged to the correct target concept? Although FIND-S will find a hypothesis consistent with the training data, it has no way to determine whether it has found the ***only*** hypothesis in ***H*** consistent with the data (i.e., the correct target concept), or whether there are many other consistent hypotheses as well.
2. Why prefer the most specific hypothesis? In case there are multiple hypotheses consistent with the training examples, FIND-S will find the most specific. It is unclear whether we should prefer this hypothesis over the most general, or some other hypothesis of intermediate generality.

Unanswered Questions of FIND-S Algorithm

3. Are the training examples consistent? In most practical learning problems there is some chance that the training examples will contain at least some errors or noise. Such inconsistent sets of training examples can severely mislead FIND-S, given the fact that it ignores negative examples. We would prefer an algorithm that could at least detect when the training data is inconsistent and, preferably, accommodate such errors.

4. What if there are several maximally specific consistent hypotheses? In the hypothesis language H for the ***EnjoySport*** task, there is always a unique, most specific hypothesis consistent with any set of positive examples. However, for other hypothesis spaces there can be several maximally specific hypotheses consistent with the data.

Consistent Hypothesis and Version Space

- ***The idea***: output a description of the set of *all hypotheses **consistent*** with the training examples (correctly classify training examples).
- ***Version Space***: a representation of the set of hypotheses which are *consistent* with *D*

  1. an explicit list of hypotheses **(List-Then-Eliminate)**
  2. a compact representation of hypotheses which exploits the *more_general_than* partial ordering **(Candidate-Elimination)**

Consistent Hypothesis and Version Space

An hypothesis h is **consistent** with a set of training examples D iff $h(x) = c(x)$ for each example in D

$$Consistent(h, D) \equiv (\forall \langle x, c(x) \rangle \in D)\ h(x) = c(x)$$

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

h1 = (?, ?, No, ?, Many) – Consistent

h2 = (?, ?, No, ?, ?) – Not Consistent

Consistent Hypothesis and Version Space

- The version space $VS_{H,D}$ is the subset of the hypothesis from H *consistent* with the training example in D

$$VS_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

List-Then-Eliminate algorithm

Version space as list of hypotheses

1. *VersionSpace* $\leftarrow$ a list containing every hypothesis in *H*
2. For each training example, $\langle x, c(x) \rangle$ Remove from *VersionSpace* any hypothesis *h* for which $h(x) \neq c(x)$
3. Output the list of hypotheses in *VersionSpace*

Consistent Hypothesis and Version Space

- F1 - > A, B
- F2 - > X, Y
- **Instance Space:** (A, X), (A, Y), (B, X), (B, Y) – **4 Examples**

- **Hypothesis Space:** (A, X), (A, Y), (A, **ø**), (A, ?), (B, X), (B, Y), (B, **ø**), (B, ?), (**ø**, X), (**ø**, Y), (**ø**, **ø**), (**ø**, ?), (**?**, X), (**?**, Y), (**?**, **ø**), (**?**, ?) - **16 Hypothesis**

- **Semantically Distinct Hypothesis :** (A, X), (A, Y), (A, ?), (B, X), (B, Y), (B, ?), (**?**, X), (**?**, Y (**?**, ?), (**ø**, **ø**) – **10**

Consistent Hypothesis and Version Space

- Version Space: (A, X), (A, Y), (A, ?), (B, X), (B, Y), (B, ?), (?, X), (?, Y) (?, ?), (ø, ø),
- Training Instances

F1	F2	Target
A	X	Yes
A	Y	Yes

- Consistent Hypothesis are: (A, ?), (?, ?)

List-Then-Eliminate algorithm

Problems

- The hypothesis space must be finite
- Enumeration of all the hypothesis, rather inefficient

Candidate Elimination Algorithm Solved Example - 1

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

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

S_0: ⟨∅, ∅, ∅, ∅, ∅. ∅⟩

S_1: ⟨*Sunny,Warm, Normal, Strong, Warm, Same*⟩

S_2: S_3: ⟨*Sunny,Warm, ?, Strong, Warm, Same*⟩

S_4 ⟨*Sunny, Warm, ?, Strong, ?, ?*⟩

G_4: ⟨*Sunny, ?, ?, ?, ?, ?*⟩ ⟨*?, Warm, ?, ?, ?, ?*⟩

G_3: ⟨*Sunny,?,?,?,?,?*⟩ ⟨*?,Warm,?,?,?,?*⟩ ⟨*?,?,Normal,?,?,?*⟩ ⟨*?, ?,?,?,Cool,?*⟩ ⟨*?,?,?,?,?,Same*⟩

G_0: G_1: G_2: ⟨?, ?, ?, ?, ?, ?⟩

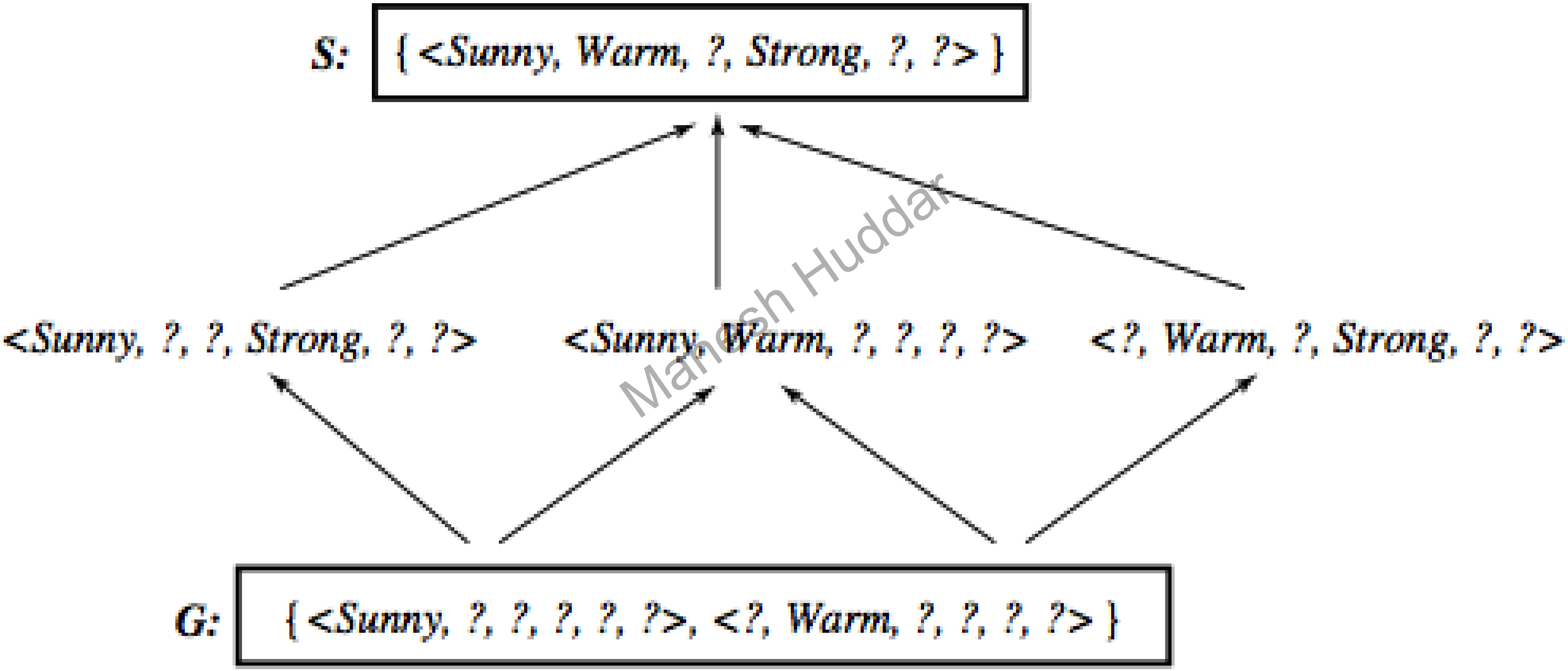
Learned Version Space by Candidate Elimination Algorithm

S $\langle$*Sunny, Warm, ?, Strong, ?, ?*$\rangle$

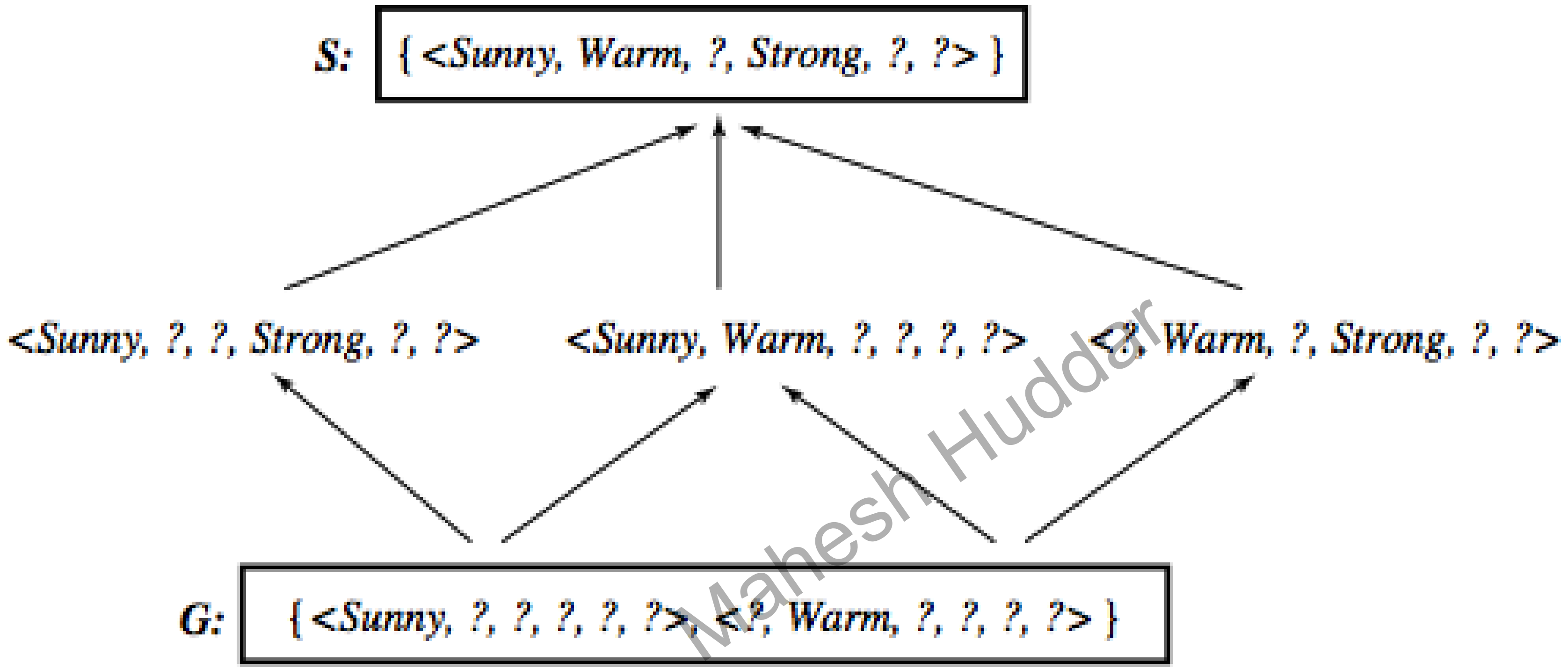
$\langle$*Sunny, ?, ?, Strong, ?, ?*$\rangle$ $\langle$*Sunny, Warm, ?, ?, ?, ?*$\rangle$ $\langle$*?, Warm, ?, Strong, ?, ?*$\rangle$

G $\langle$*Sunny, ?, ?, ?, ?, ?*$\rangle$ $\langle$*?, Warm, ?, ?, ?, ?*$\rangle$

Learned Version Space by Candidate Elimination Algorithm



New instances to be classified



Instance	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	Light	Warm	Same	?
C	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

Candidate Elimination Algorithm Solved Example - 2

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Candidate Elimination Algorithm Solved Example - 2

S0: (0, 0, 0)

S1: (0, 0, 0)

S2: (0, 0, 0)

S3: (Small, Red, Circle)

S4: (Small, Red, Circle)

S5: (Small, ?, Circle)

S: G: (Small, ?, Circle)

G5: (Small, ?, Circle)

G4: (Small, ?, Circle)

G3: (Small, ?, Circle)

G2: (Small, Blue, ?)   (Small, ?, Circle)   (?, Blue, ?)   (Big, ?, Triangle)   (?, Blue, Triangle)

G1: (Small, ?, ?)   (?, Blue, ?)   (?, ?, Triangle)

G0: (?, ?, ?)

SIze	Color	Shape	Class / Label
Big	Red	Circle	No
Small	Red	Triangle	No
Small	Red	Circle	Yes
Big	Blue	Circle	No
Small	Blue	Circle	Yes

Candidate Elimination Algorithm Solved Example - 3

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Candidate Elimination Algorithm Solved Example - 3

example	citations	size	inLibrary	price	editions	buy
1	some	small	no	affordable	many	no
2	many	big	no	expensive	one	yes
3	some	big	always	expensive	few	no
4	many	medium	no	expensive	many	yes
5	many	small	no	affordable	many	yes

- S0: (0, 0, 0, 0, 0)
- S1: (0, 0, 0, 0, 0)
- S2: (Many, Big, No, Exp, One)
- S3: (Many, Big, No, Exp, One)
- S4: (Many, ?, No, Exp, ?)
- **S5: (Many, ?, No, ?, ?)**
- (Many, ?, No, ?, ?)
- **G5: (Many, ?, ?, ?, ?)**
- G4: (Many,?,?,?,?) (Many,?,?,Exp,?) (?,?,No,exp,?)
- G3: (Many,?,?,?,?) (Many, big,?,?,?) (?,Big,no,?,?) (?,Big,?,Aff,?) (?,Big,?,?,Many) (?,Big,?,?,One) (Many,?,?,Exp,?) (?,Small,?,Exp,?) (?,Medium,?,Exp,?) (?,?,No,exp,?) (?,?,?,Exp,one) (?,?,?,Exp,many) (?,?,?,?,One)
- G2: (Many,?,?,?, ?) (?, Big,?,?,?) (?,?,?,Exp,?) (?,?,?,?,One)
- G1: (Many,?,?,?, ?) (?, Big,?,?,?) (?,Medium,?,?,?) (?,?,Always,?,?) (?,?,?,Exp,?) (?,?,?,?,One) (?,?,?,?,Few)
- G0: (?, ?, ?, ?, ?)

Final Hypothesis Set: (Many, ?, No, ?, ?) (Many, ?, ?, ?, ?)

Candidate Elimination Algorithm Solved Example - 4

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Candidate Elimination Algorithm - Solved Example - 4

Example	Shape	Size	Color	Surface	Thickness	Target Concept
1	Circular	Large	Light	Smooth	Thick	Malignant (+)
2	Circular	Large	Light	Irregular	Thick	Malignant (+)
3	Oval	Large	Dark	Smooth	Thin	Benign (-)
4	Oval	Large	Light	Irregular	Thick	Malignant (+)

S0: (0, 0, 0, 0, 0)

S1: (Circular, Large, Light, Smooth, Thick)

S2: (Circular, Large, Light, ?, Thick)

S3: (Circular, Large, Light, ?, Thick)

S4: (?, Large, Light, ?, Thick)

G4: (?, ?, Light, ?, ?) (?, ?, ?, ?, Thick)

G3: (Circular, ?, ?, ?, ?) (?, ?, Light, ?, ?) (?, ?, ?, ?, Thick)

G2: (?, ?, ?, ?, ?)

G1: (?, ?, ?, ?, ?)

G0: (?, ?, ?, ?, ?)

Candidate Elimination Algorithm Solved Example - 5

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Candidate Elimination Algorithm - Solved Example - 5

S0: (0, 0, 0, 0, 0)

S1: (Round, Triangle, Round, Purple, Yes)

S2: (Round, Triangle, Round, Purple, Yes)

S3: (?, Triangle, Round, ?, Yes)

S4: (?, Triangle, Round, ?, Yes)

S5: (?, ?, Round, ?, Yes)

G5: (?, ?, Round, ?, Yes)

Ex	Eyes	Nose	Head	Fcolor	Hair	Smile
1	Round	Triangle	Round	Purple	Yes	Yes
2	Square	Square	Square	Green	Yes	No
3	Square	Triangle	Round	Yellow	Yes	Yes
4	Round	Triangle	Round	Green	No	No
5	Square	Square	Round	Yellow	Yes	Yes

G4: (Square, Triangle, ?, ?, ?) (?, Triangle, Square, ?, ?) (?, Triangle, ?, Yellow, ?) (?, Triangle, ?, Purple, ?) (?, Triangle, ?, ?, yes)
(Square, ?, Round, ?, ?) (?, Square, Round, ?, ?) (?, ?, Round, Yellow, ?) (?, ?, Round, Purple, ?) (?, ?, Round, ?, Yes)

G3: (?, Triangle, ?, ?, ?) (?, ?, Round, ?, ?)

G2: (Round, ?, ?, ?, ?) (?, Triangle, ?, ?, ?) (?, ?, Round, ?, ?) (?, ?, ?, Purple, ?)

G1: (?, ?, ?, ?, ?)

G0: (?, ?, ?, ?, ?)

Candidate Elimination Algorithm - Solved Example - 5

Ex	Eyes	Nose	Head	Fcolor	Hair	Smile
1	Round	Triangle	Round	Purple	Yes	Yes
2	Square	Square	Square	Green	Yes	No
3	Square	Triangle	Round	Yellow	Yes	Yes
4	Round	Triangle	Round	Green	No	No
5	Square	Square	Round	Yellow	Yes	Yes

S0: (0, 0, 0, 0, 0)

S1: (Round, Triangle, Round, Purple, Yes)

S2: (Round, Triangle, Round, Purple, Yes)

S3: (?, Triangle, Round, ?, Yes)

S4: (?, Triangle, Round, ?, Yes)

S5: (?, ?, Round, ?, Yes)

G5: (?, ?, Round, ?, Yes)

G4: (Square, Triangle, ?, ?, ?) (?, Triangle, Square, ?, ?) (?, Triangle, ?, Yellow, ?) (?, Triangle, ?, Purple, ?) (?, Triangle, ?, ?, yes)

(Square, ?, Round, ?, ?) (?, Square, Round, ?, ?) (?, ?, Round, Yellow, ?) (?, ?, Round, Purple, ?) (?, ?, Round, ?, Yes)

G3: (?, Triangle, ?, ?, ?) (?, ?, Round, ?, ?)

G2: (Round, ?, ?, ?, ?) (?, Triangle, ?, ?, ?) (?, ?, Round, ?, ?) (?, ?, ?, Purple, ?)

G1: (?, ?, ?, ?, ?)

G0: (?, ?, ?, ?, ?)

Candidate Elimination Algorithm

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Candidate elimination algorithm

For each training example d, do:

$If\ d$ is *positive* example

Remove from G any hypothesis h inconsistent with d

For each hypothesis s in S not consistent with d:

- Remove s from S
- Add to S all minimal generalizations of s consistent with d and having a generalization in G
- Remove from S any hypothesis with a more specific h in S

$If\ d$ is *negative* example

Remove from S any hypothesis h inconsistent with d

For each hypothesis g in G not consistent with d:

- Remove g from G
- Add to G all minimal specializations of g consistent with d and having a specialization in S
- Remove from G any hypothesis having a more general hypothesis in G

Remarks On Version Spaces And Candidate-Elimination Algorithm

1. Will the CANDIDATE-ELIMINATION Algorithm Converge to the Correct Hypothesis?
2. How Can Partially Learned Concepts Be Used?
3. What Training Example Should the Learner Request Next?

Remarks On Version Spaces And Candidate-Elimination Algorithm

Will the CANDIDATE-ELIMINATION Algorithm Converge to the Correct Hypothesis?

- The learned Version Space correctly describes the target concept, provided:
    1. There are no errors in the training examples
    2. There is some hypothesis that correctly describes the target concept
- If S and G converge to a single hypothesis the concept is exactly learned
- An empty version space means no hypothesis in H is consistent with training examples

Remarks On Version Spaces And Candidate-Elimination Algorithm

How Can Partially Learned Concepts Be Used?

- The learner is required to classify new instances that it has not yet observed.
- Even though the version space of Figure contains multiple hypotheses, indicating that the target concept has not yet been fully learned, it is possible to classify certain examples with the same degree of confidence as if the target concept had been uniquely identified.

Remarks On Version Spaces And Candidate-Elimination Algorithm

How Can Partially Learned Concepts Be Used?

- Note that although instance A was not among the training examples, it is classified as a positive instance by ***every*** hypothesis in the current version space (shown in Figure**).**
- Because the hypotheses in the version space unanimously agree that this is a positive instance, the learner can classify instance A as positive with the same confidence it would have if it had already converged to the single, correct target concept.
- Regardless of which hypothesis in the version space is eventually found to be the correct target concept, it is already clear that it will classify instance A as a positive example.

Remarks On Version Spaces And Candidate-Elimination Algorithm

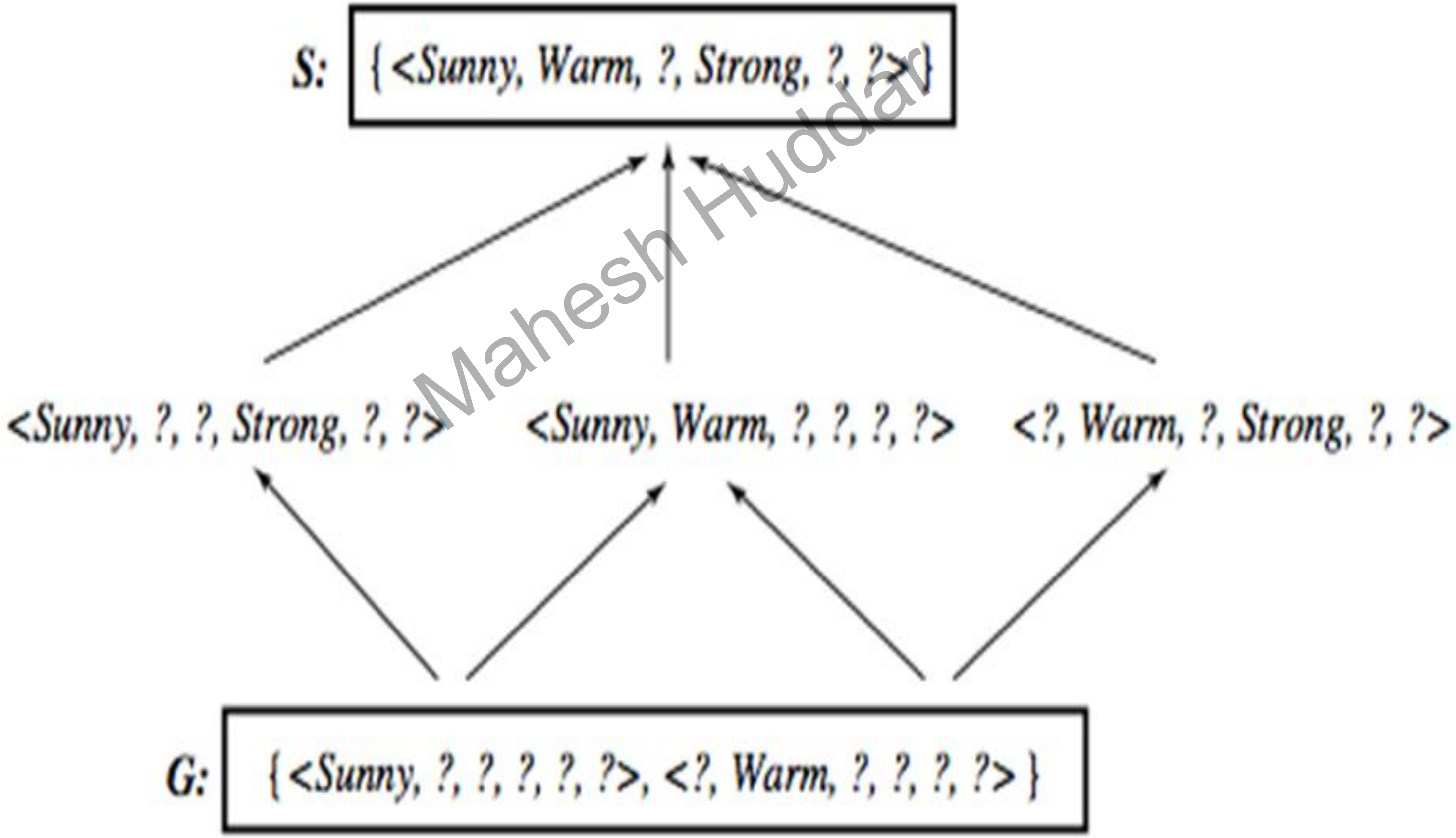
How Can Partially Learned Concepts Be Used?

- Similarly, instance B is classified as a negative instance by every hypothesis in the version space. This instance can therefore be safely classified as negative, given the partially learned concept.
- Instance C presents a different situation. Half of the version space hypotheses classify it as positive and half classify it as negative. Thus, the learner cannot classify this example with confidence until further training examples are available.
- Finally, instance D is classified as positive by two of the version space hypotheses and negative by the other four hypotheses. In this case we have less confidence in the classification than in the unambiguous cases of instances A and B. Still, the vote is in favor of a negative classification, and one approach we could take would be to output the majority vote, perhaps with a confidence rating indicating how close the vote was.

Remarks On Version Spaces And Candidate-Elimination Algorithm

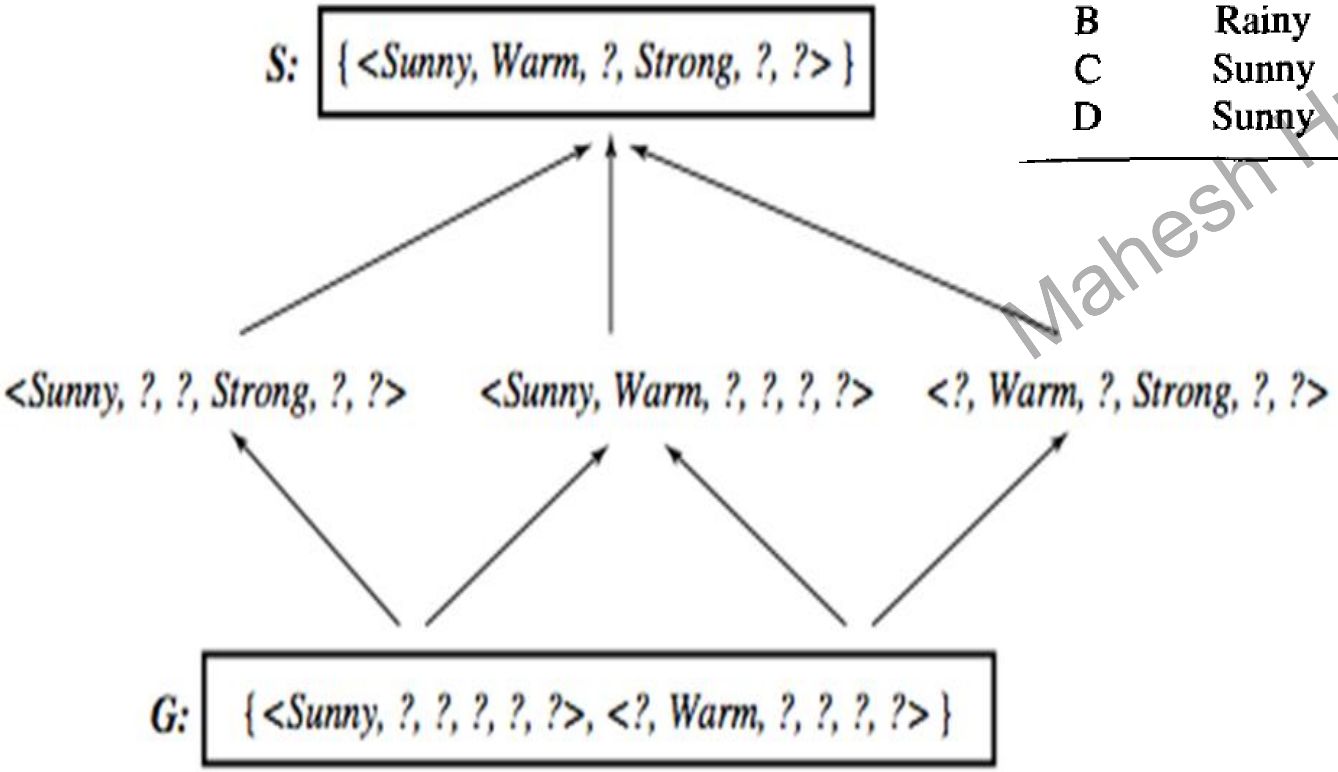
How Can Partially Learned Concepts Be Used?

- The learner is required to classify new instances that it has not yet observed.



Remarks On Version Spaces And Candidate-Elimination Algorithm

How Can Partially Learned Concepts Be Used?



Instance	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
A	Sunny	Warm	Normal	Strong	Cool	Change	?
B	Rainy	Cold	Normal	Light	Warm	Same	?
C	Sunny	Warm	Normal	Light	Warm	Same	?
D	Sunny	Cold	Normal	Strong	Warm	Same	?

Remarks On Version Spaces And Candidate-Elimination Algorithm

What Training Example Should the Learner Request Next?

- Up to this point we have assumed that training examples are provided to the learner by some external teacher.
- Suppose instead that the learner is allowed to conduct experiments in which it chooses the next instance, then obtains the correct classification for this instance from an external nature.

Remarks On Version Spaces And Candidate-Elimination Algorithm

What Training Example Should the Learner Request Next?

- Consider the version space learned from the four training examples of the Enjoysport concept.
- What would be a good query for the learner to pose at this point?
- What is a good query strategy in general?
- The learner should choose an instance that would be classified positive by some of these hypotheses, but negative by others.

Unbiased Learner

Candidate Elimination Algorithm

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Inductive Bias of Candidate-Elimination Algorithm

- The CANDIDATE-ELIMINATION Algorithm will converge toward the true target concept provided
  - it is given accurate training examples and
  - its initial hypothesis space contains the target concept.
- What if the target concept is not contained in the hypothesis space?
- Can we avoid this difficulty by using a hypothesis space that includes every possible hypothesis?

Inductive Bias of Candidate-Elimination Algorithm

A Biased Hypothesis Space

- Suppose we wish to assure that the hypothesis space contains the unknown target concept.
- The obvious solution is to enrich the hypothesis space to include ***every possible*** hypothesis.
- To illustrate, consider the ***EnjoySport*** example in which we restricted the hypothesis space to include only conjunctions of attribute values.

<Sunny, High, Normal, Strong, Cool, Same>

Inductive Bias of Candidate-Elimination Algorithm

A Biased Hypothesis Space

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

S0: <∅, ∅, ∅, ∅, ∅. ∅>

S1: <Sunny, Warm, Normal, Strong, Cool, Change>

S2: <?, Warm, Normal, Strong, Cool, Change>

S3: <?, Warm, Normal, Strong, Cool, Change>

G2: <?, ?, ?, ?, ?, ?>

G1: <?, ?, ?, ?, ?, ?>

G0: <?, ?, ?, ?, ?, ?>

Inductive Bias of Candidate-Elimination Algorithm

A Biased Hypothesis Space

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Cool	Change	Yes
2	Cloudy	Warm	Normal	Strong	Cool	Change	Yes
3	Rainy	Warm	Normal	Strong	Cool	Change	No

- Because of the restriction, the hypothesis space is unable to represent even simple disjunctive target concepts such as ***"Sky*** = ***Sunny*** or ***Sky*** = ***Cloudy."***
- **< *"Sky* = *Sunny* or *Sky* = *Cloudy"*, Warm, Normal, Strong, Cool, Change>**

Inductive Bias of Candidate-Elimination Algorithm

Unbiased Learner

- The obvious solution to the problem of assuring that the target concept is in the hypothesis space H is to provide a hypothesis space capable of representing ***every teachable concept;*** that is, it is capable of representing every possible subset of the instances X.
- In general, the set of all subsets of a set X is called ***the powerset*** of X.

Inductive Bias of Candidate-Elimination Algorithm

Unbiased Learner

- In the EnjoySport learning task, for example, the size of the instance space X of days described by the six available attributes is 96.
- $|H'| = 2^{|X|}$, or 2^{96} (vs $|H| = 973$, a strong bias)
- **< “*Sky* = *Sunny*** or ***Sky* = *Cloudy*”, ?, ?, ?, ?>**

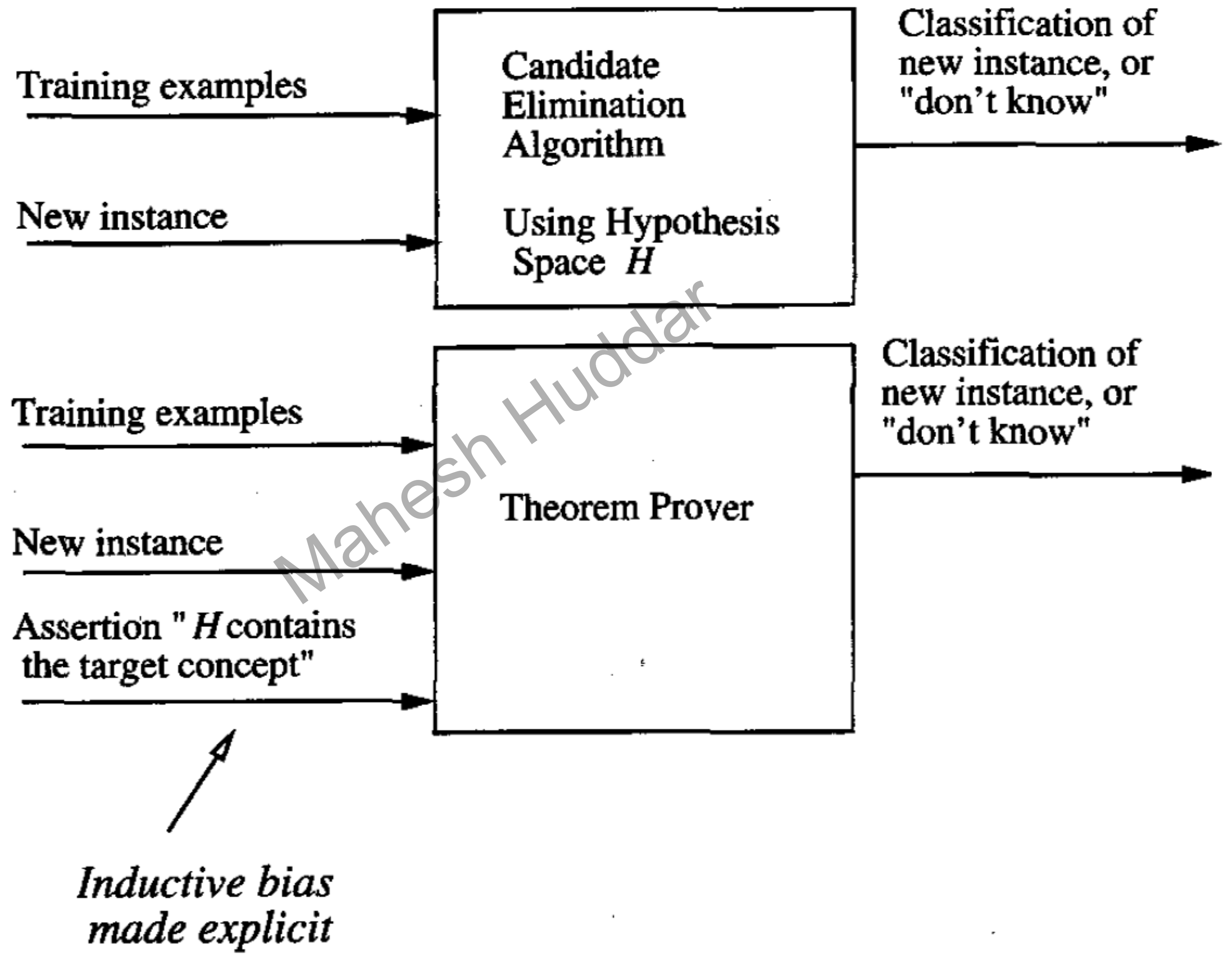
  $\langle Sunny, ?, ?, ?, ?, ?\rangle \vee <Cloudy, ?, ?, ?, ?, ?\rangle$
- We are guaranteed that the target concept exists

Inductive Bias of Candidate-Elimination Algorithm

Inductive System

Equivalent

Deductive System



The Inductive Learning Hypothesis

The Inductive Learning Hypothesis

Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Issues in Machine Learning

1. What algorithms exist for learning general target functions from specific training examples? In what settings will particular algorithms converge to the desired function, given sufficient training data? **Which algorithms perform best for which types of problems and representations?**
2. **How much training data is sufficient?** What general bounds can be found to relate the confidence in learned hypotheses to the amount of training experience and the character of the learner's hypothesis space?
3. When and how can **prior knowledge** held by the learner guide the process of generalizing from examples? Can prior knowledge be helpful even when it is only approximately correct?

Issues in Machine Learning

4. What is the best strategy for choosing a useful next training experience, and how does the choice of this strategy alter the complexity of the learning problem?
5. What is the best way to reduce the learning task to one or more function approximation problems? Put another way, what specific functions should the system attempt to learn? Can this process itself be automated?
6. How can the learner automatically alter **its** representation to improve its ability to represent and learn the target function?