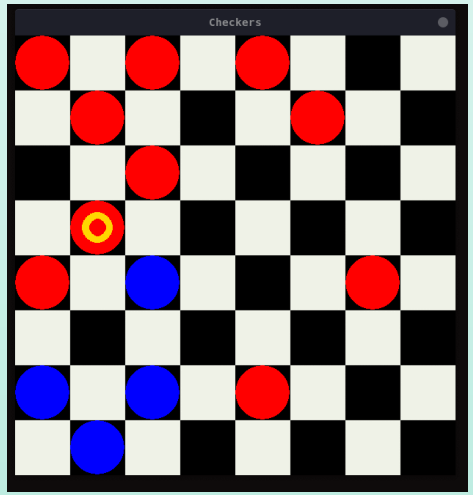
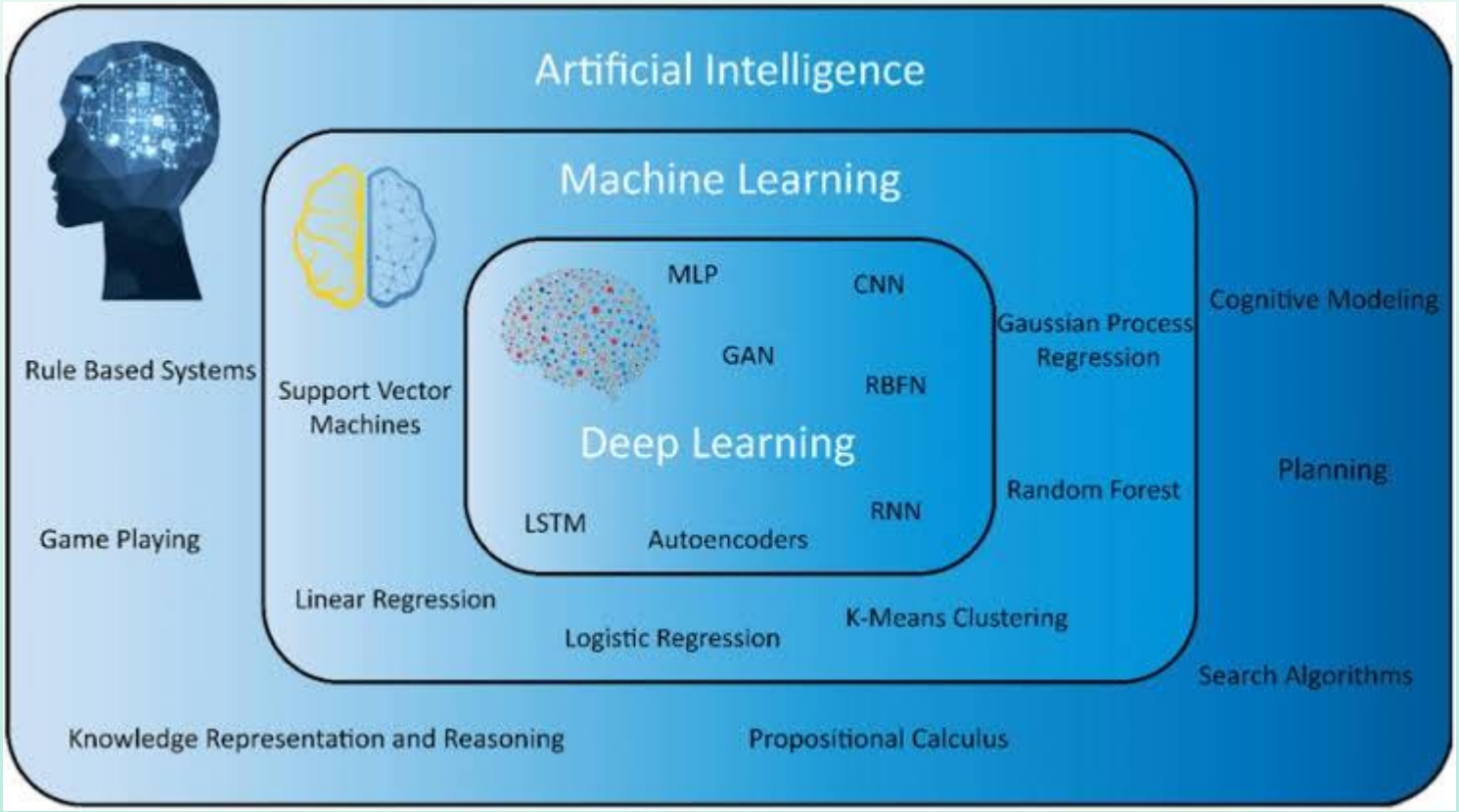
- The term “machine learning” was coined by Arthur Samuel, a computer scientist at IBM and a pioneer in AI and computer gaming. Samuel designed a computer program for playing checkers. The more the program played, the more it learned from experience, using algorithms to make predictions.



RELATIONSHIP BETWEEN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

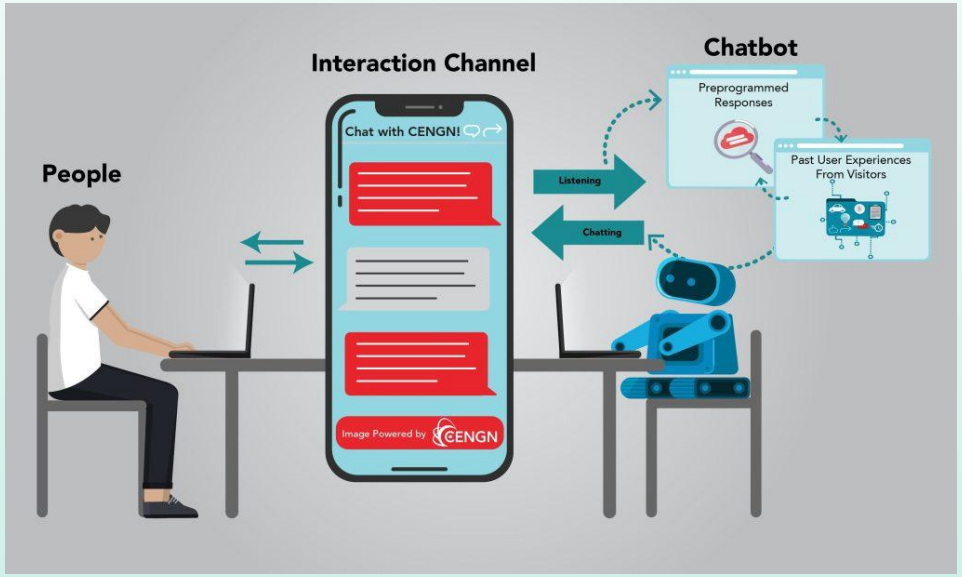


WHAT IS MACHINE LEARNING?

- Computers learn from data.
- Machine learning is a subset of AI, which uses algorithms that learn from data to make predictions.
- Machine learning (ML) is a branch of artificial intelligence (AI) and computer science that focuses on the using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy.
- Machine learning is programming computers to optimize a performance criterion using example data or past experience.
- The capability of a machine to imitate intelligent human behavior.
- Enables computers to learn from data and make decisions or predictions without being explicitly programmed to do so.

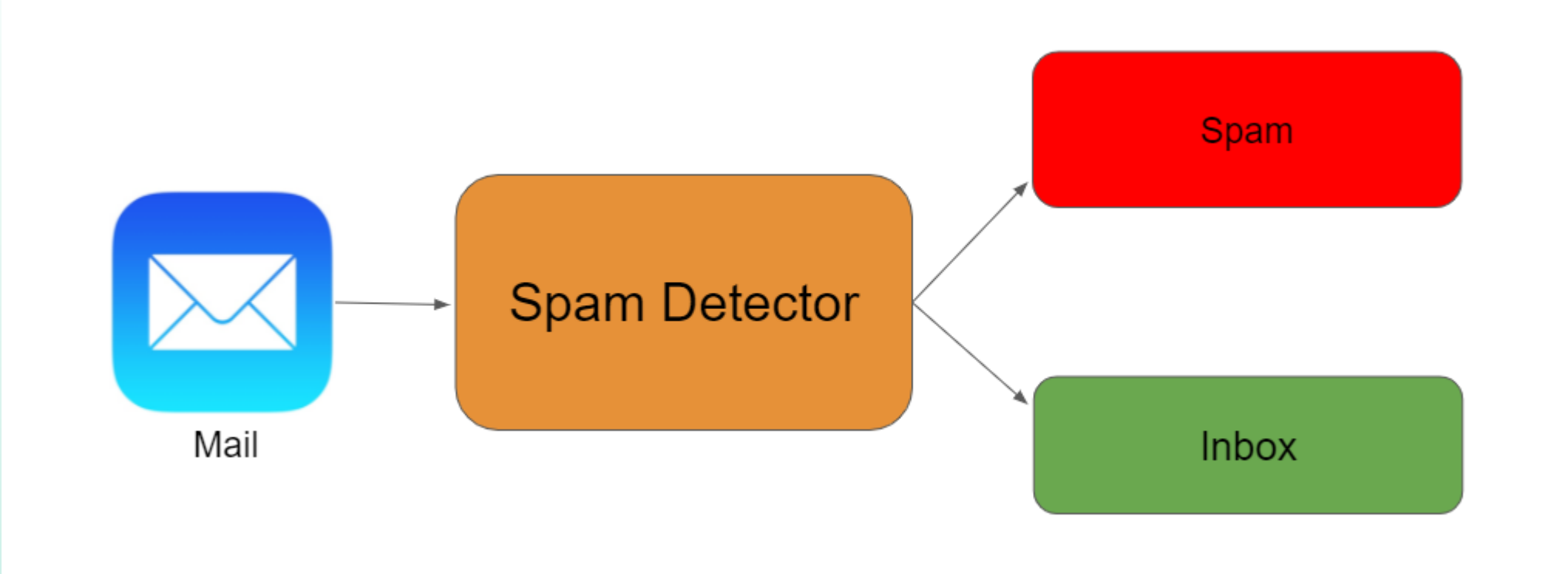
ADVANTAGES OF MACHINE LEARNING

1. Chatbot



Chatbots are conversational artificial intelligence systems trained using machine learning to provide the appropriate response or assistance based on inputs. These systems learn from past experiences, such as questions asked by previous visitors/responses given and from datasets containing possible future queries/appropriate answers.

2. SPAM DETECTION



The ML works best when it comes to implementing spam detection. Earlier, e-mail service providers were using rule-based approaches for spam filtering. ML has replaced it where phishing emails and spam messages can be detected using ANN(Artificial Neural Network), including AI techniques.

3. PRODUCT RECOMMENDATIONS

Most brands, especially eCommerce brands, are using ML for product recommendations. For training the system, the customer's purchase history and inventory are used for identifying the hidden pattern. When customers log in to their accounts, they get the product recommendation, motivating them to purchase the items. In turn, it increases sales.

4. STOCK MARKET PREDICTIONS

These machine learning methods help predict how the stock market will perform based on year-to-year analysis. Using predictive analytics machine learning models, analysts can predict the stock price for 2025 and beyond.

5. TRAFFIC PREDICTIONS

When you use Google Maps to map your commute to work or a new restaurant in town, it provides an estimated time of arrival. Google uses machine learning to build models of how long trips will take based on historical traffic data (gleaned from satellites). It then takes that data based on your current trip and traffic levels to predict the best route according to these factors.

6. IMPROVED CYBER SECURITY

Several cyber threats exist, and it is a major concern of all businesses to keep data safe from cyber-attacks. One of the best real-time examples is- Microsoft Windows Defender Advanced Threat Protection.

7. SELF-DRIVING CAR TECHNOLOGY

A frequently used type of machine learning is reinforcement learning, which is used to power self-driving car technology. Self-driving vehicle company Waymo uses machine learning sensors to collect data of the car's surrounding environment in real time. This data helps guide the car's response in different situations, whether it is a human crossing the street, a red light, or another car on the highway..

Real World Applications of Machine Learning

Face Recognition

Siri and Cortana

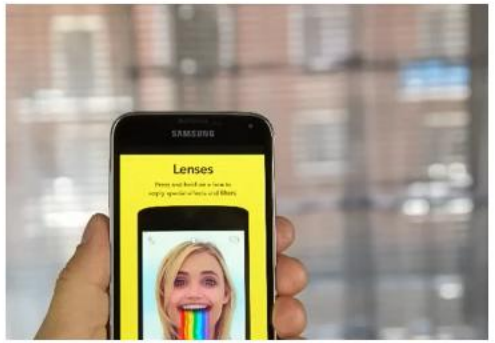
Healthcare Industry

Weather Forecasting

Produce a Web Series

Machine Learning Example

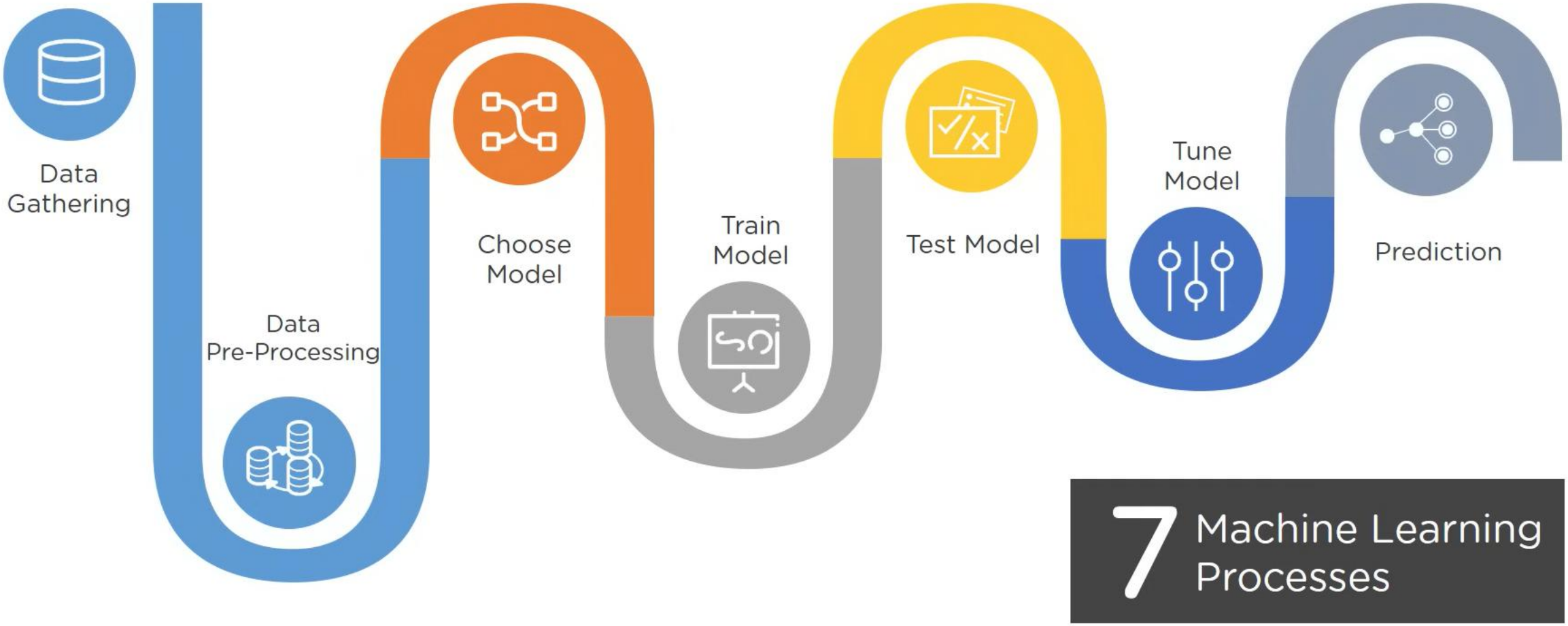
Matrix of numbers



Looks for color patterns on face

Points align themselves and look for areas of contrast

Processes involved in Machine Learning



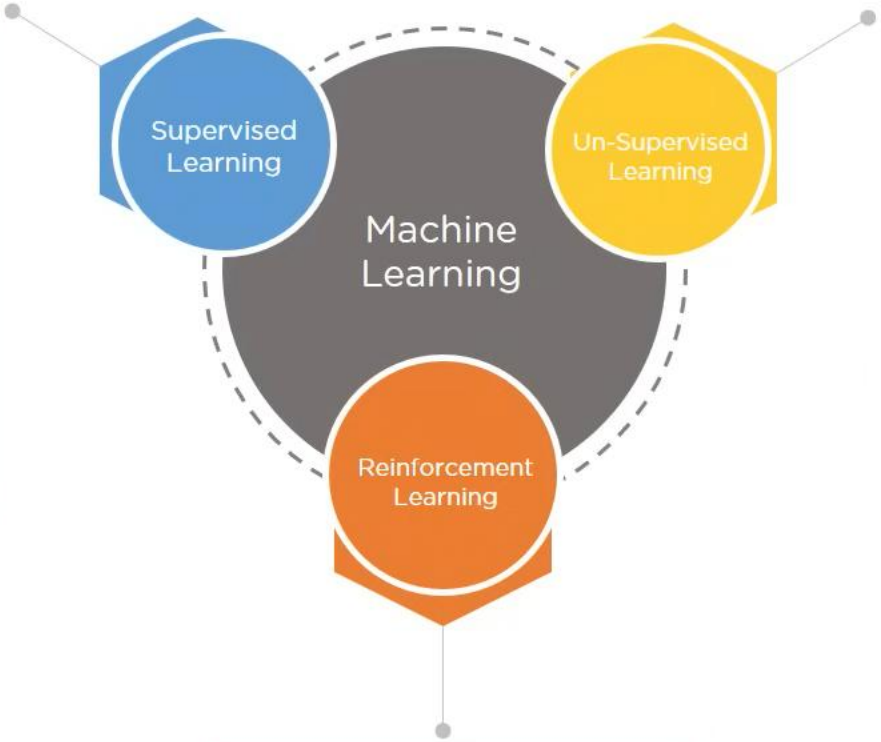
Types of Machine Learning Algorithms

Classification

- Fraud Detection
- Email Spam Detection
- Image Classification

Regression

- Weather Forecasting
- Risk Assessment
- Score Prediction



Association

- Market Basket Analysis
- Text Mining
- Face Recognition

Clustering

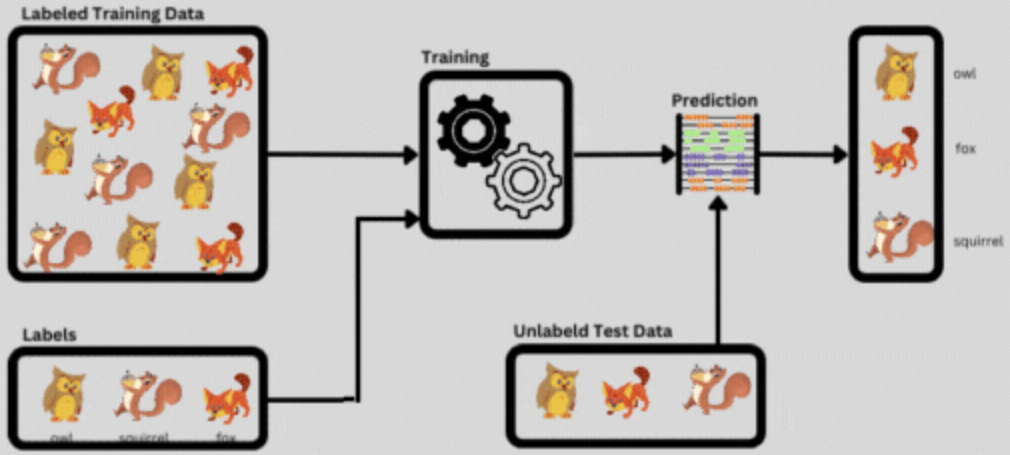
- Medical Research
- City Planning
- Targeted Marketing

Reinforcement Learning

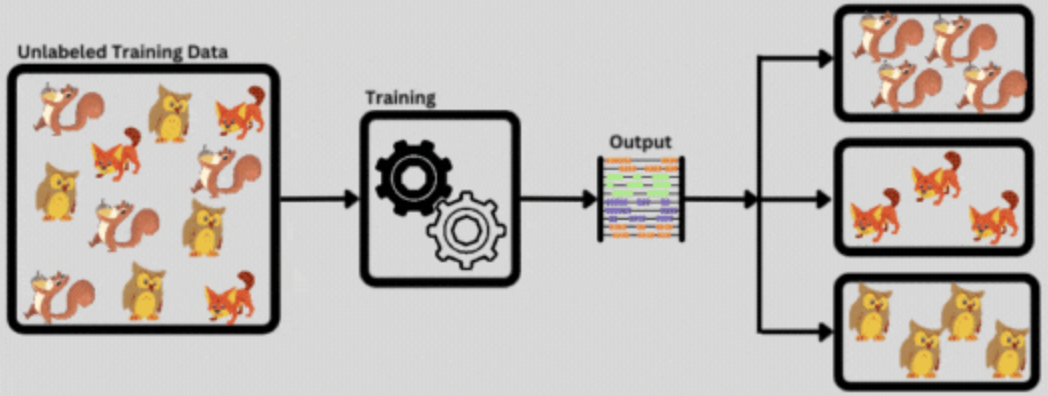
- Gaming
- Robot Navigation
- Stock Trading
- Assembly Line Processes

Machine Learning

Supervised Learning

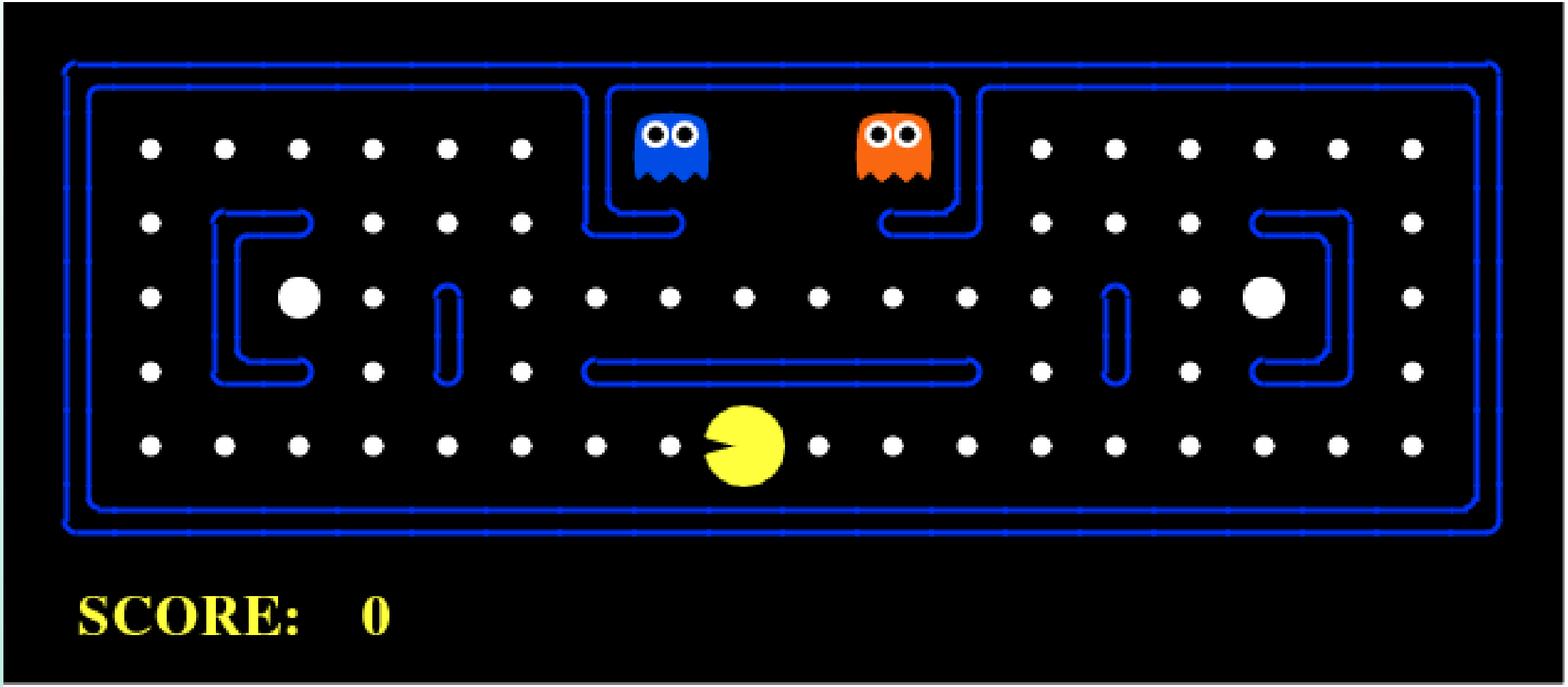


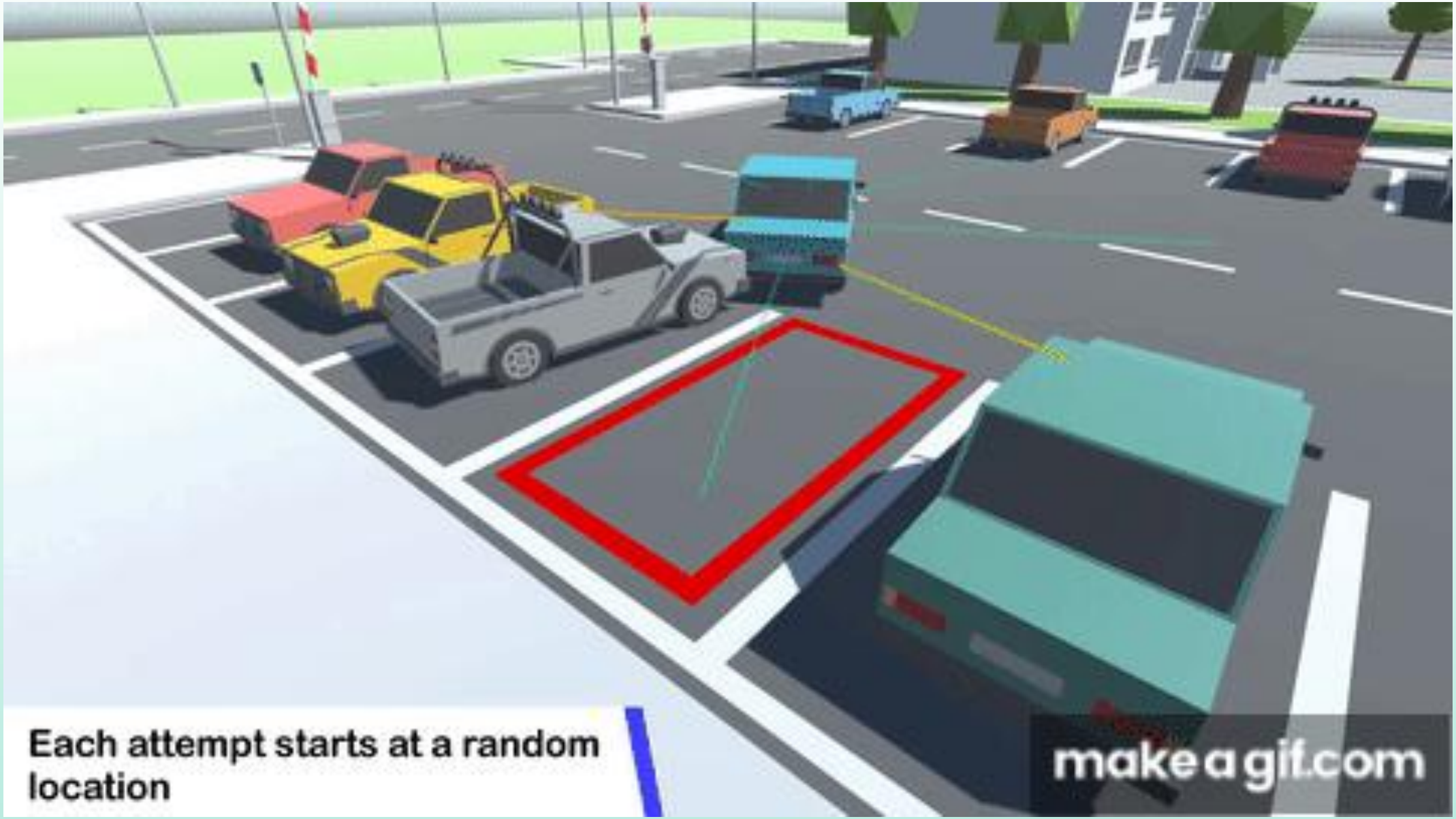
Unsupervised Learning



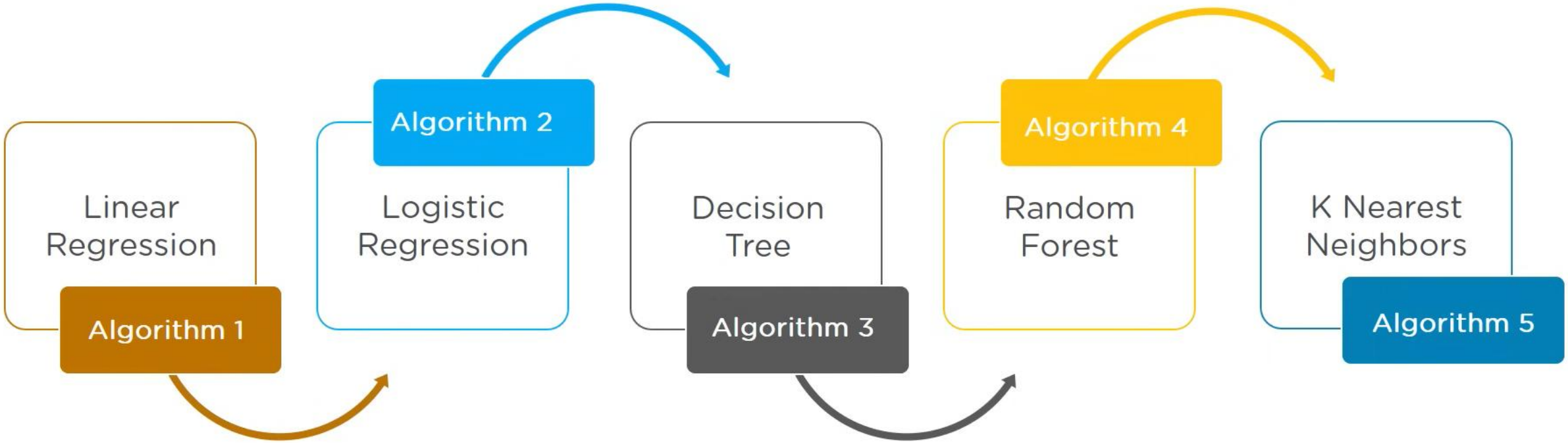
Datamapu

Reinforcement Learning Example



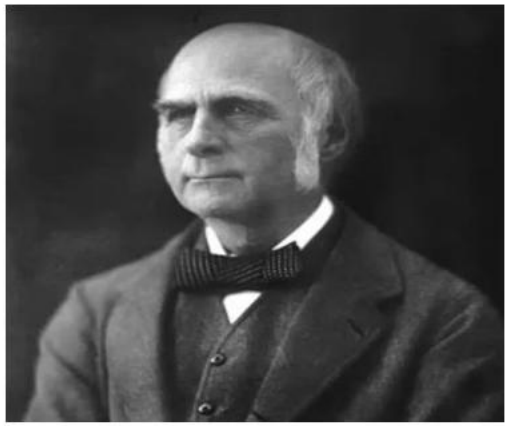

Each attempt starts at a random location
makeagif.com

Popular Algorithms in Machine Learning



History of Linear Regression

Linear Regression: Brief history on how Regression came into picture.

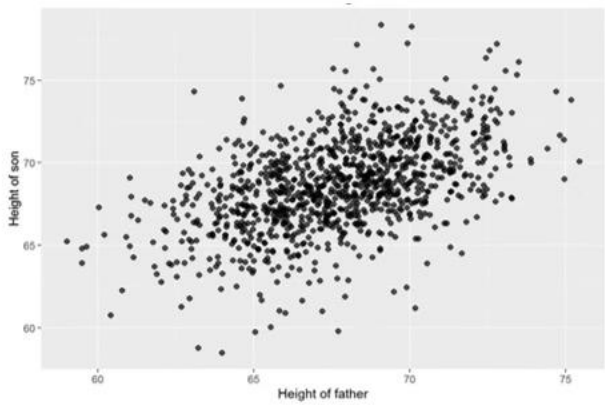


Francis Galton

Studied

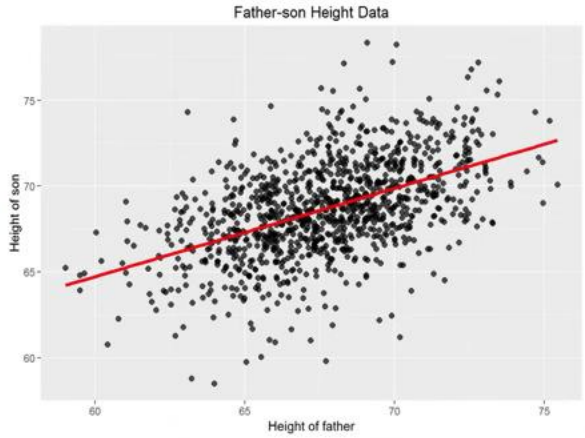
Heights of Father and Son

Analyzed



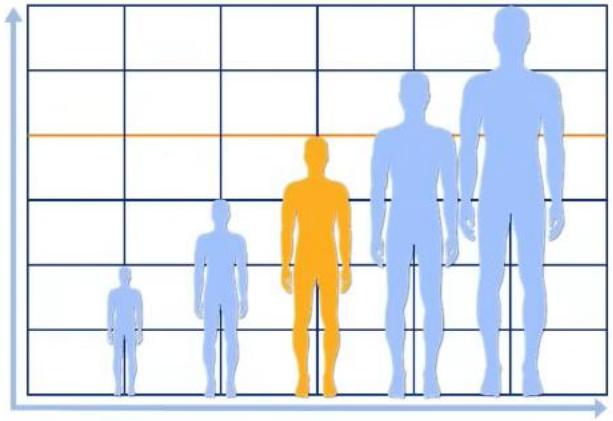
Height Data

Best fit line



Regression Line

Regressed

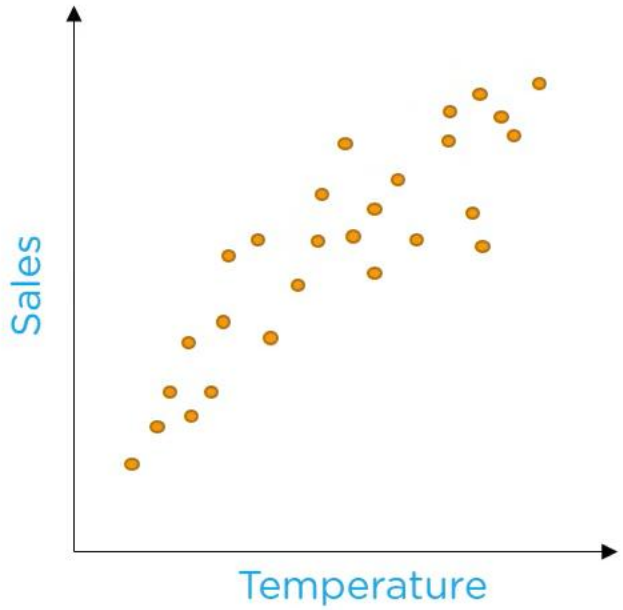


Mean Height of all People

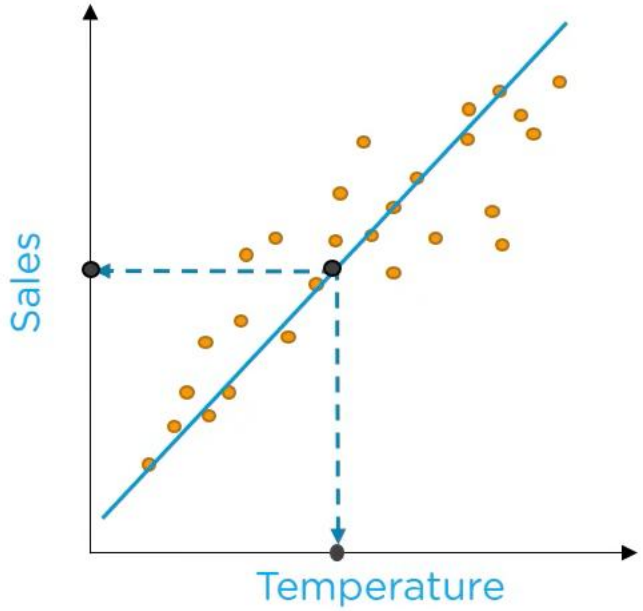
Linear Regression

Linear Regression is a linear modelling approach to find relationship between one or more independent variables (predictors) denoted as **X** and dependent variable (target) denoted as **Y**.

Predict sales of Ice Cream based on temperature:



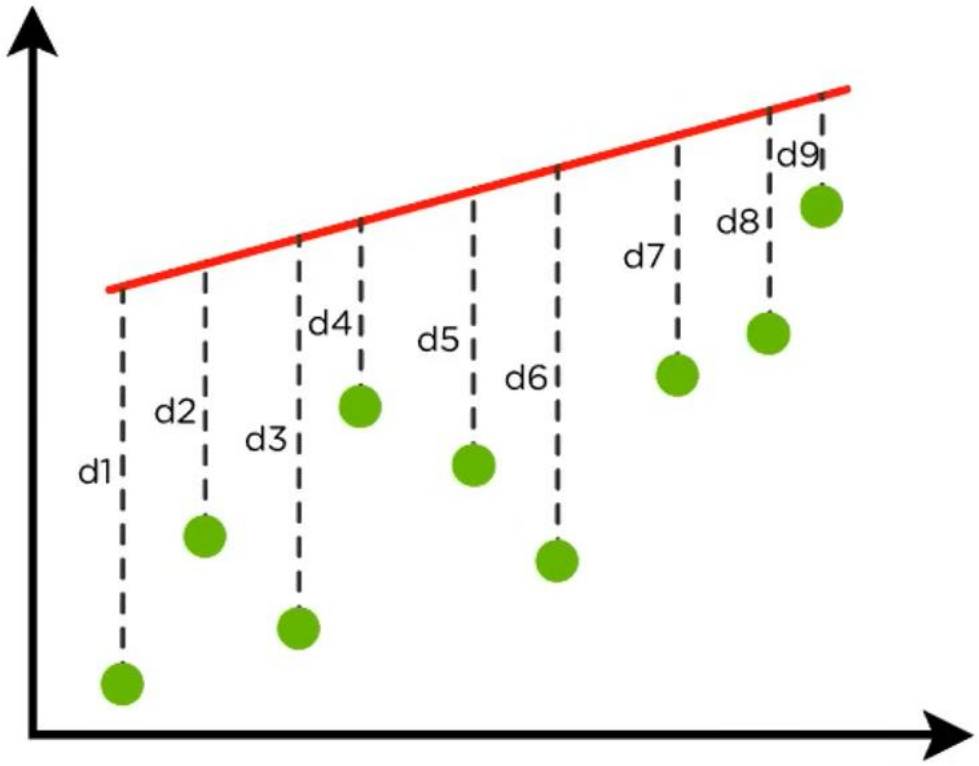
Plotting the sales of ice cream based on temperature



Regression line to predict the sales

Linear Regression

Finding the best fit line: The best fit line can be found out by minimizing the distance between all the data points and the distance to the regression line. Ways to minimize this distance are sum of squared errors, sum of absolute errors etc.



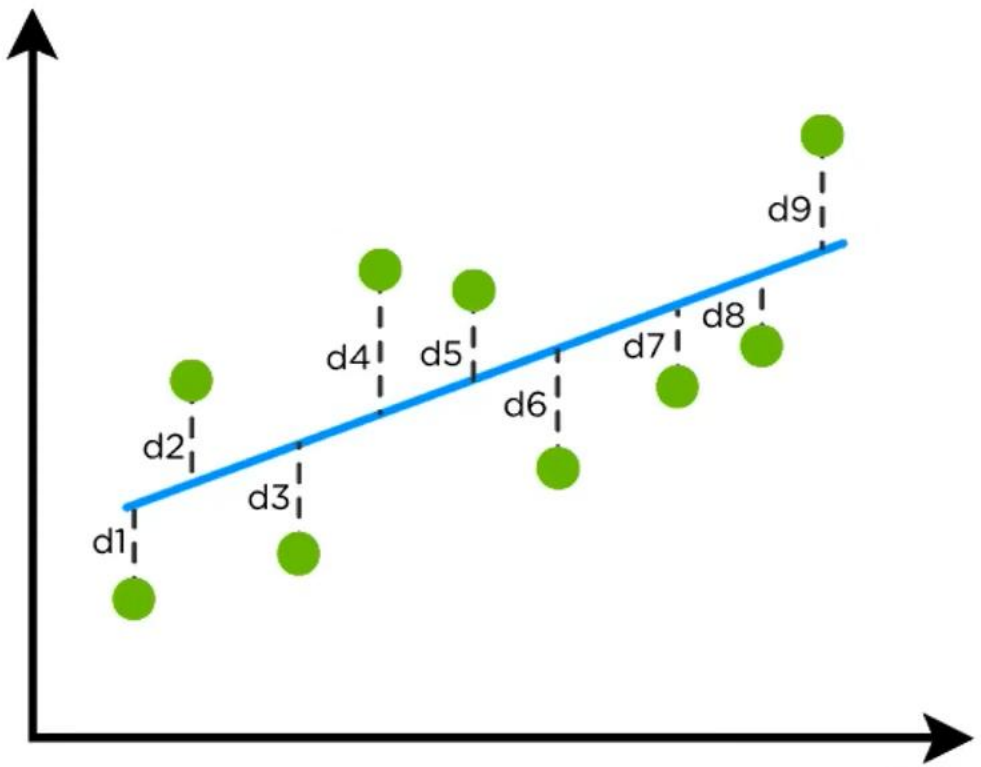
$$D1 = d1^2 + d2^2 + d3^2 + d4^2 + d5^2 + d6^2 + d7^2 + d8^2 + d9^2$$

Not the best fit line

Distance between the data points and the line is maximum

Linear Regression

Finding the best fit line: The best fit line can be found out by minimizing the distance between all the data points and the distance to the regression line. Ways to minimize this distance are sum of squared errors, sum of absolute errors etc.



Best fit regression line

Distance between the data points and the line is the least

$$D = d1^2 + d2^2 + d3^2 + d4^2 + d5^2 + d6^2 + d7^2 + d8^2 + d9^2$$

The regression line (blue) has the least value of D

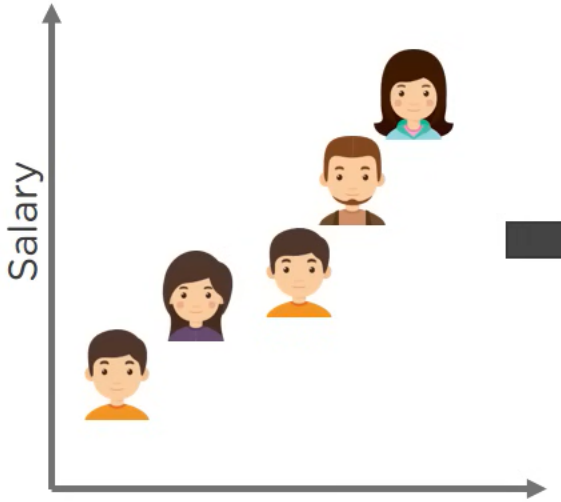
Implementation of Linear Regression

Linear Regression: Predict Employee Salary based on Years of Experience.

New Joinee

How much do we pay her?

HR team trying to figure out how much to pay to a new joinee?



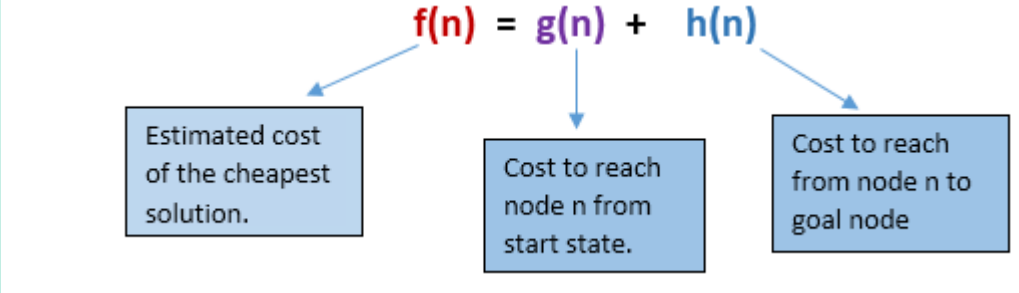
Employee data of Salary based on Years of experience

Predict Salary

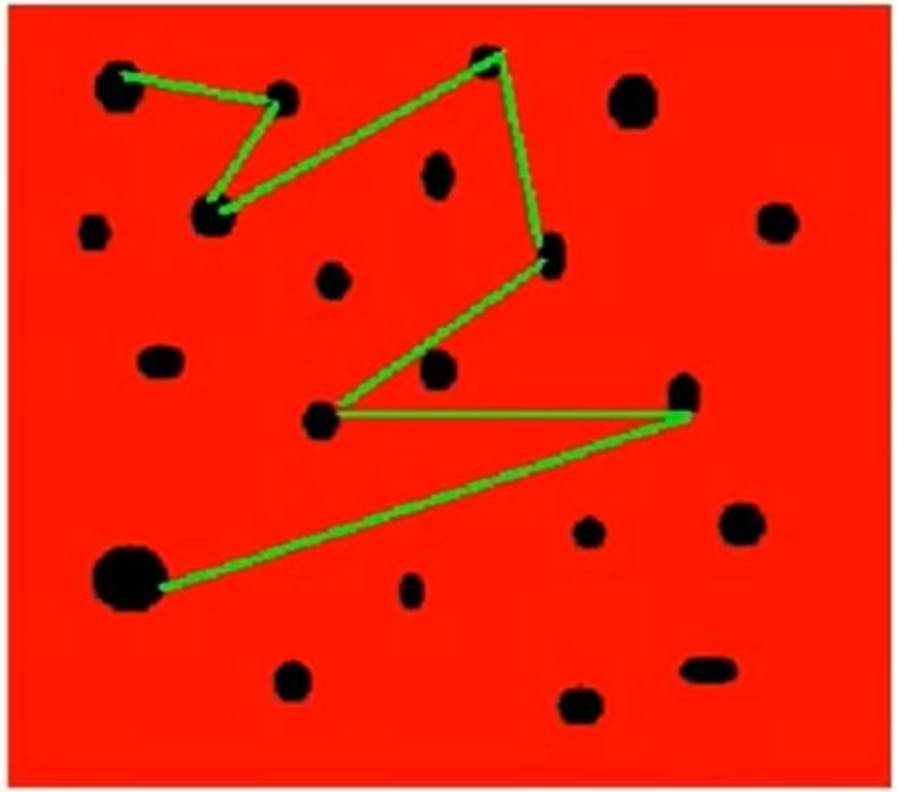
A* Algorithm

A* search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster.

In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a **fitness number**.



CONCEPT LEARNING



What is Concept Learning?

Concept Space

Hypothesis Space

Distinct Hypothesis Space

Machine Learning

Concept Learning in Machine Learning

- The problem of inducing general functions from specific training examples is central to concept 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.

Tom enjoys his favorite water sports

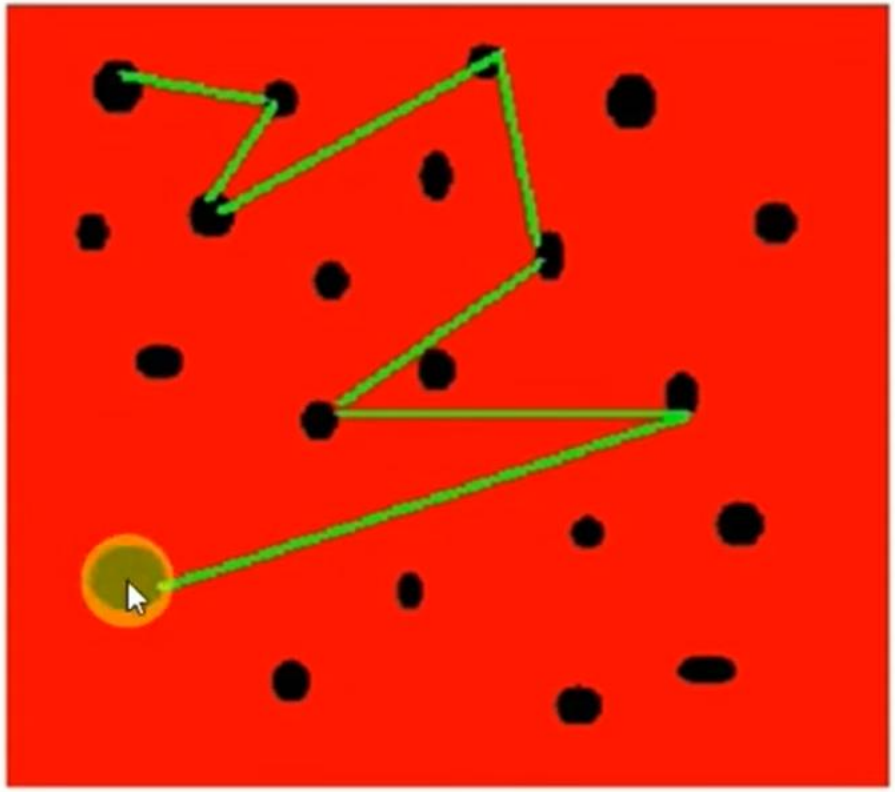
Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	*EnjoySport*
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

understanding and recognizing patterns or ideas (concepts) from examples, allowing you to apply that understanding to new situations.

Concept Learning in Machine Learning

What is Concept Learning…?

- "A task of acquiring potential hypothesis (solution) that best fits the given training examples."



Concept Learning in Machine Learning

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

- *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*).

Concept Learning in Machine Learning

- What hypothesis representation shall we provide to the learner in this case?
- 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.

Concept Learning in Machine Learning

- F1 - > A, B
- F2 - > X, Y

? → Accept Anything
ø → Dose not accept

- **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**

Every hypothesis containing one or more " Φ" symbols represents the empty set of instances; that is, it classifies every instance as *negative*.

The instance space is the set of all possible observations or examples

hypothesis space is the set of all possible rules or models that a learning algorithm can represent.

A semantically distinct hypothesis means that two hypotheses are considered different if they represent different concepts, even if they might look similar syntactically.

Concept Learning in Machine Learning

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

Concept Learning in Machine Learning

Hypothesis Space: A set of all possible hypotheses

Possible syntactically distinct Hypotheses for *EnjoySport*

= 5 * 4 * 4 * 4 * 4 * 4 = 5120 973

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

Find S Algorithm

- The find-S algorithm is a basic concept learning algorithm in machine learning.
- The find-S algorithm finds the most specific hypothesis that fits all the positive examples.
- We have to note here that the algorithm considers only those positive training example.
- 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.

Important Representation :

1.? indicates that any value is acceptable for the attribute.
2.specify a single required value (e.g., Cold) for the attribute.
3.ϕindicates that no value is acceptable.
4.The most **general hypothesis** is represented by: **{?, ?, ?, ?, ?, ?}**
5.The most **specific hypothesis** is represented by: **{ϕ, ϕ, ϕ, ϕ, ϕ, ϕ}**

Algorithm

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

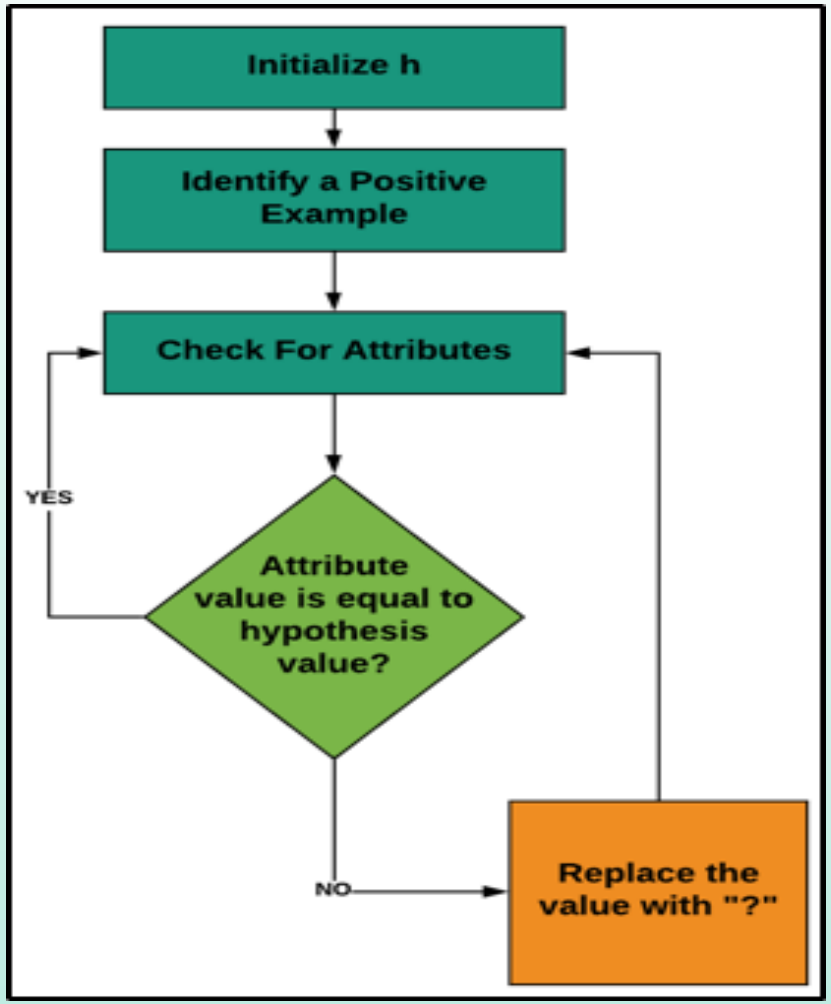
   For each attribute constraint a, in h

   If the constraint a, is satisfied by x

   Then do nothing

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

How Does It Work?



1.The process starts with initializing ‘h’ with the most specific hypothesis, generally, it is the first positive example in the data set.
2.We check for each positive example. If the example is negative, we will move on to the next example but if it is a positive example we will consider it for the next step.
3.We will check if each attribute in the example is equal to the hypothesis value.
4.If the value matches, then no changes are made.
5.If the value does not match, the value is changed to ‘?’.
6.We do this until we reach the last positive example in the data set.

Limitations of Find-S Algorithm

- There is no way to determine if the hypothesis is consistent throughout the data.
- Inconsistent training sets can actually mislead the Find-S algorithm, since it ignores the negative examples.
- Find-S algorithm does not provide a backtracking technique to determine the best possible changes that could be done to improve the resulting hypothesis.

The attribute EnjoySport indicates whether or not a Person 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 ?

Implementation of Find-S Algorithm

Which days does someone come out to enjoy 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

1. Initialize h to the most specific hypothesis in H

h0 = <Ø, Ø, Ø, Ø, Ø, Ø>

- **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), "∅" (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.

TABLE The *EnjoySport* concept learning task.

Step 2 of Find-S Algorithm First iteration

h0 = (ø, ø, ø, ø, ø, ø, ø)

X1 = <Sunny, Warm, Normal, Strong, Warm, Same>

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

Step 2 of Find-S Algorithm Second iteration

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

X2 = <Sunny, Warm, High, Strong, Warm, Same>

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

Step 2 of Find-S Algorithm Third iteration

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

X3 = <Rainy, Cold, High, Strong, Warm, Change> – No

X3 is Negative example Hence ignored

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

Step 2 of Find-S Algorithm Fourth iteration

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

X4 = <Sunny, Warm, High, Strong, Cool, Change>

h4 = <Sunny, Warm, ?, Strong, ?, ?>

Step 3

The final maximally specific hypothesis is **<Sunny, Warm, ?, Strong, ?, ?>**

Implementation of Find-S Algorithm

what days does a person likes to go on walk?

Time	Weather	Temperature	Company	Humidity	Wind	Goes
Morning	Sunny	Warm	Yes	Mild	Strong	Yes
Evening	Rainy	Cold	No	Mild	Normal	No
Morning	Sunny	Moderate	Yes	Normal	Normal	Yes
Evening	Sunny	Cold	Yes	High	Strong	Yes

Implementation of Find-S Algorithm

which particular seeds are poisonous ?

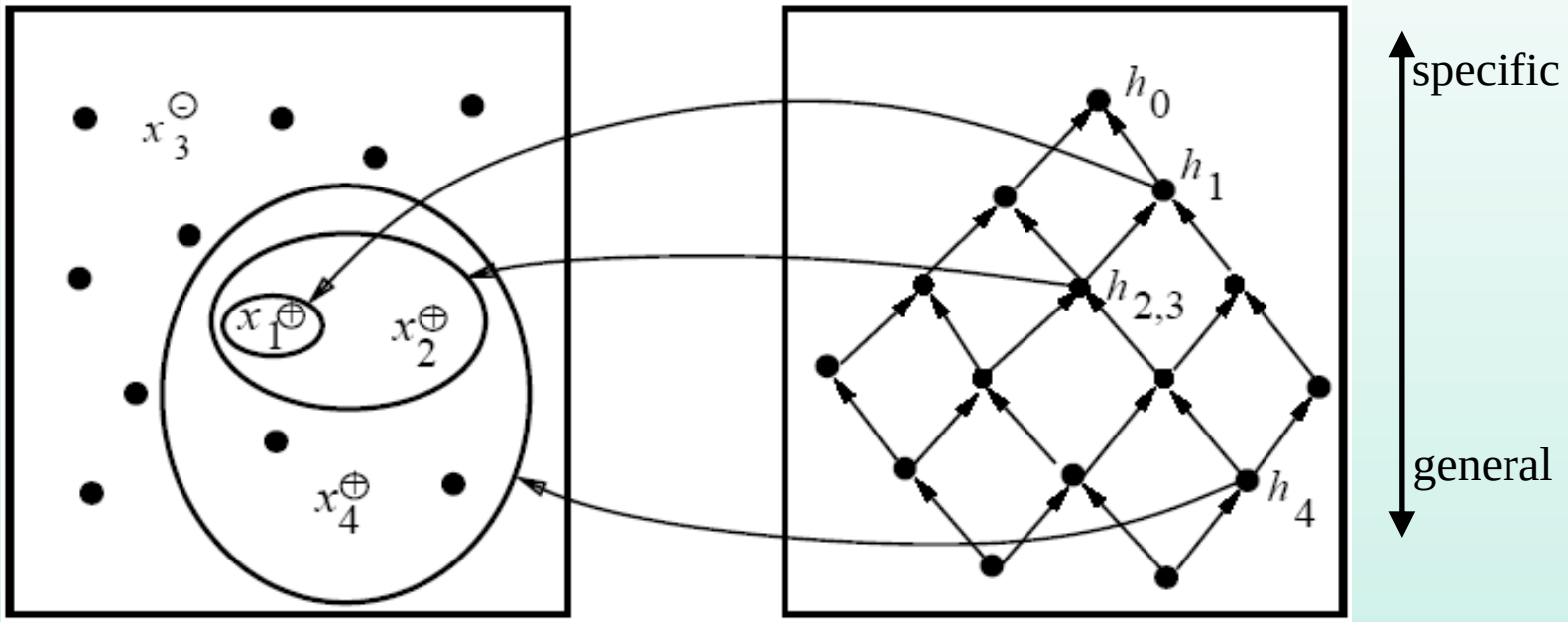
EXAMPLE	COLOR	TOUGHNESS	FUNGUS	APPEARANCE	POISONOUS
1.	GREEN	HARD	NO	WRINKELD	YES
2.	GREEN	HARD	YES	SMOOTH	NO
3.	BROWN	SOFT	NO	WRINKLED	NO
4.	ORANGE	HARD	NO	WRINKLED	YES
5.	GREEN	SOFT	YES	SMOOTH	YES
6.	GREEN	HARD	YES	WRINKLED	YES
7.	ORANGE	HARD	NO	WRINKLED	YES

ATTRIBUTES ON WHICH THE CONCEPT DEPENDS ON

CONCEPT

EXAMPLE OF FIND-S

Instances X Hypotheses H



x_1=⟨*Sunny Warm Normal Strong Warm Same*⟩ +

x_2=⟨*Sunny Warm High Strong Warm Same*⟩ +

x_3=⟨*Rainy Cold High Strong Warm Change*⟩ –

x_4=⟨*Sunny Warm High Strong Cool Change*⟩ +

h_0=⟨∅ ∅ ∅ ∅ ∅ ∅⟩

h_1=⟨*Sunny Warm Normal Strong Warm Same*⟩

h_2=⟨*Sunny Warm ? Strong Warm Same*⟩

h_3=⟨*Sunny Warm ? Strong Warm Same*⟩

h_4=⟨*Sunny Warm ? Strong ? ?*⟩

WHAT IS PANDAS?

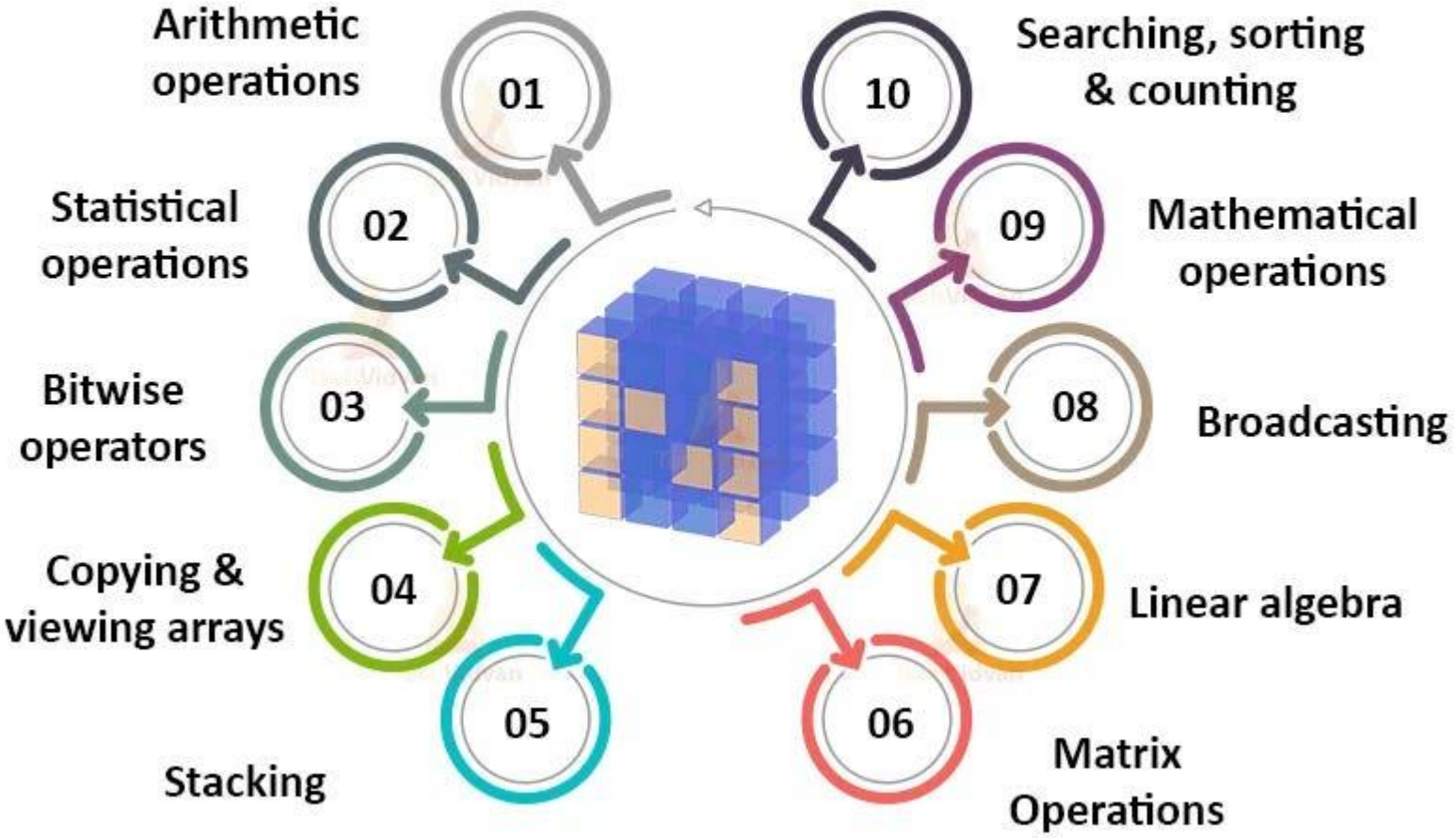
Pandas is one of the most used libraries in Python for data science or data analysis. It can read data from CSV or Excel files, manipulate the data, and generate insights from it. Pandas can also be used to clean data, filter data, and visualize data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008

Function	Description
Pandas read_csv() Function	This function is used to retrieve data from CSV files in the form of a dataframe.
Pandas head() Function	This function is used to return the top n (5 by default) values of a data frame or series.
Pandas tail() Function	This method is used to return the bottom n (5 by default) rows of a data frame or series.
Pandas unique() Function	It returns all the unique values in a particular column.
Pandas nunique() Function	Returns the number of unique values in the column
Pandas isnull() Function	Returns the DataFrame/Series of the boolean values. Missing values gets mapped to True and non-missing value gets mapped to False.

WHAT IS NUMPY?

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.

Uses of NumPy



Enumerate() Function Adds A Counter To Each Item In A List Or Other Iterable

```
a = ["java", "python", "c"]

# Iterating list using enumerate to get both index and element
for i, name in enumerate(a):
    print(f"Index {i}: {name}")

# Converting to a list of tuples
print(list(enumerate(a)))

a = ["java", "python", "c"]

#Looping through the list using enumerate
# starting the index from 1
for index, x in enumerate(a, start=1):
    print(index, x)
print(list(enumerate(a, start=1)))
```

```
Index 0: java
Index 1: python
Index 2: c
[(0, 'java'), (1, 'python'), (2, 'c')]

1 java
2 python
3 c
[(1, 'java'), (2, 'python'), (3, 'c')]
```

```
def find_s(training_data):
hypothesis = None

    for example in training_data:
        features, target = example[:-1], example[-1]  # Split attributes and target
        if target == "Yes":  # Consider only positive examples
            if hypothesis is None:
                hypothesis = features  # First positive example sets the hypothesis
            else:
                # Generalize hypothesis where necessary
                hypothesis = [
                    h if h == f else "?" for h, f in zip(hypothesis, features)
                ]

    return hypothesis
```

```
# Example training data: (Sky, Temperature, Humidity, Wind, Water, Forecast, EnjoySport)
training_examples = [
    ["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"]
]

# Run Find-S algorithm
hypothesis = find_s(training_examples)
print("Final Hypothesis:", hypothesis)
```


Final Hypothesis: ['Sunny', 'Warm', '?', 'Strong', '?', '?']

PROBLEMS WITH FIND-S

- Problems:
  - Throws away information!
    - Negative examples
  - Can’t tell whether it has learned the concept
    - Depending on *H*, there might be several h’s that fit
    - Picks a maximally specific
  - Can’t tell when training data is inconsistent
- But
  - It is simple
  - Outcome is independent of order of examples
- What alternative overcomes these problems?
  - Keep *all* consistent hypotheses!
    - Candidate elimination algorithm

CONSISTENT HYPOTHESES AND VERSION SPACE

$h(x) = c(x)$ => hypothesis **h** correctly classifies the training example x according to the target concept c.

- A hypothesis h is consistent with a set of training examples D of target concept c
  if $h(x) = c(x)$ for each training example $\langle x, c(x)\rangle$ in D
  - Note that consistency is with respect to specific D.
- Notation:

$$consistent(h, d) \equiv \forall \langle x, c(x)\rangle \in d :: h(x) = c(x)$$

- The version space, $vs_{h,d}$, with respect to hypothesis space h and training examples d, is the subset of hypotheses from h consistent with d
- Notation:

$$VS_{H,D} = \{h \,/\, h \in H \wedge consistent(h, D)\}$$

REPRESENTING VERSION SPACES

- Want more compact representation of VS
  - Store most/least general boundaries of space
  - Generate all intermediate h's in VS
  - Idea that any h in VS must be consistent with all te's
    - Generalize from most specific boundaries
    - Specialize from most general boundaries
- The general boundary, G, of version space $VS_{H,D}$ is the set of its maximally general members consistent with D
  - Summarizes the negative examples
- The specific boundary, S, of version space $VS_{H,D}$ is the set of its maximally specific members consistent with D
  - Summarizes the positive examples

Theorem

Every member of the version space lies between the S,G boundary

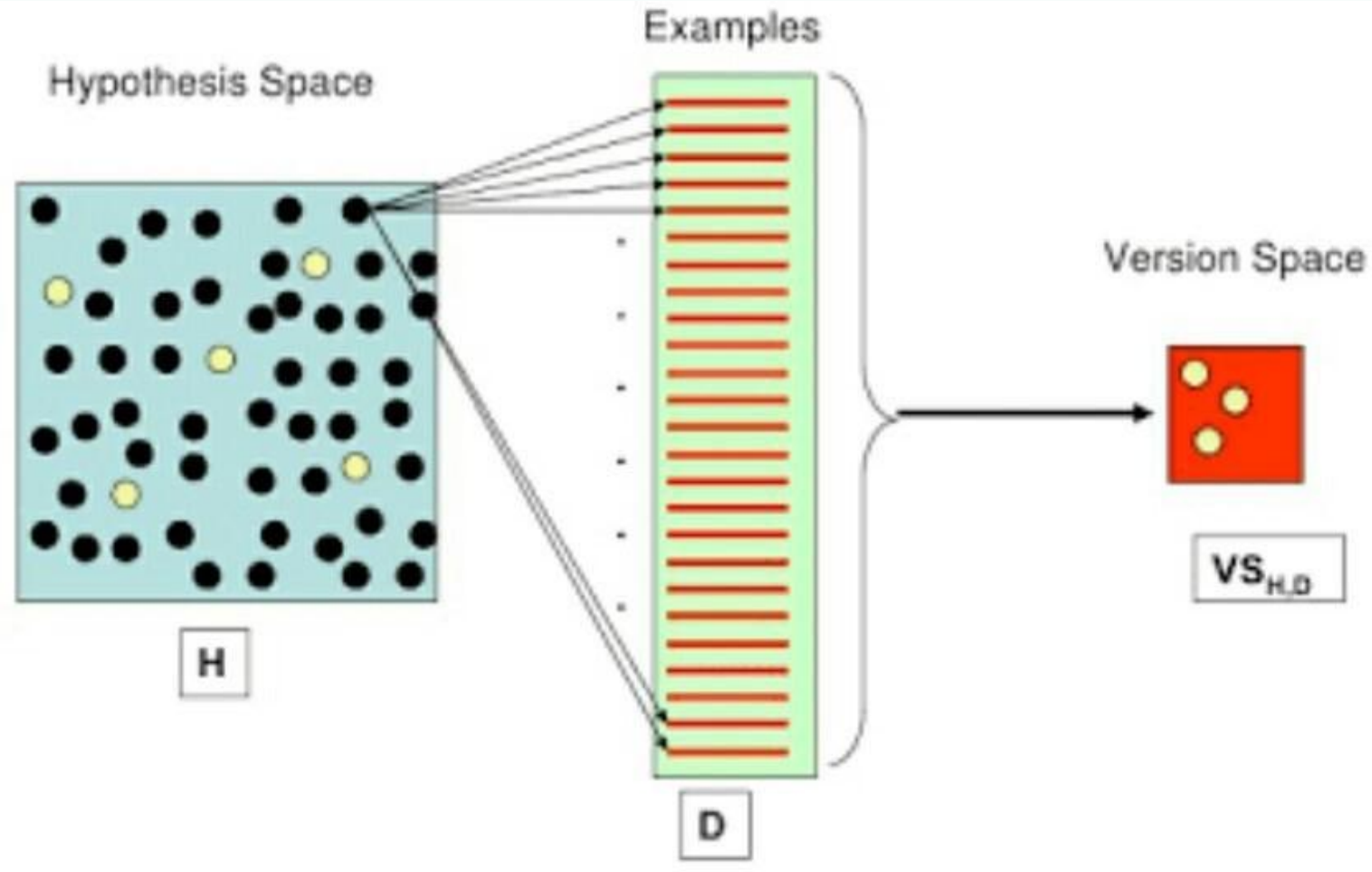
$$VS_{H,D} = \{h \mid h \in H \wedge \exists s \in s\ \exists g \in g\ (g \geq h \geq s)\}$$

- Must prove:
  - 1) every h satisfying RHS is in $VS_{H,D;}$
  - 2) every member of $VS_{H,D}$ satisfies RHS.
- For 1), let g, h, s be arbitrary members of g, h, s respectively with g>h>s
  - S must be satisfied by all + tes and so must h because it is more general;
  - G cannot be satisfied by any – tes, and so nor can h
  - H is in $VS_{H,D}$ since satisfied by all + tes and no – tes
- For 2),
  - Since h satisfies all + tes and no – tes, h ≥ s, and $g \geq h$.

Version Space The version space is **the subspace of the hypothesis space which is consistent with the** training examples

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

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


Examples
Hypothesis Space
Version Space
H
D
$VS_{H,D}$

THE LIST-THEN-ELIMINATE ALGORITHM

The LIST-THEN-ELIMINATE algorithm first initializes the version space to contain all hypotheses in H and then eliminates any hypothesis found inconsistent with any training example.

THE LIST-THEN-ELIMINATE ALGORITHM

1. ***VersionSpace c*** *a* list containing every hypothesis in H
2. For each training example, ***(x, c(x))***

   remove from ***VersionSpace*** any hypothesis ***h*** for which ***h(x) ≠ c(x)***
3. Output the list of hypotheses in ***VersionSpace***

The LIST-THEN-ELIMINATE Algorithm

- **List-Then-Eliminate** works in principle, so long as version space is finite.
- However, since it requires exhaustive enumeration of all hypotheses in practice it is not feasible.

Candidate Elimination Algorithm

$G \leftarrow$ maximally general hypotheses in H

$S \leftarrow$ maximally specific hypotheses in H

For each training example d, do

- If d is positive
  - Remove from G every hypothesis inconsistent with d
  - For each hypothesis s in S that is inconsistent with d
    - Remove s from S
    - Add to S all minimal generalizations h of s such that

      1. H is consistent with d, and

      2. Some member of G is more general than h
  - Remove from S every hypothesis that is more general than another hypothesis in S

Contd..

- If *d* is a negative example
  - Remove from *S* every hypothesis inconsistent with *d*
  - For each hypothesis *g* in *G* that is inconsistent with *d*
    - Remove *g* from *G*
    - Add to *G* all minimal specializations *h* of *g* such that
      1. *H* is consistent with *d*, and
      2. Some member of *S* is more specific than *h*
  - Remove from *G* every hypothesis that is less general than another hypothesis in *G*
- Essentially use
  - Postives to generalize S
  - Negatives to specialize G
- Convergence guaranteed if:
  - ***No errors***
  - ***There is h in H describing c***.

Implementation

Which days does someone come out to enjoy 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

AN ILLUSTRATIVE EXAMPLE

The boundary sets are first initialized to G_o and S_o, the most general and most specific hypotheses in H.

S_0 $\langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle$

G_0 $\langle ?, ?, ?, ?, ?, ? \rangle$

For training example d,

$\langle$Sunny, Warm, Normal, Strong, Warm, Same $\rangle$ +

S_0 $\langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing. \varnothing \rangle$

S_1 $\langle Sunny, Warm, Normal, Strong, Warm, Same \rangle$

G_0, G_1 $\langle ?, ?, ?, ?, ?, ? \rangle$

For training example d,

$\langle$Sunny, Warm, High, Strong, Warm, Same$\rangle$ +

S_1 $\langle$*SUNNY, WARM, NORMAL, STRONG, WARM, SAME*$\rangle$

S_2 $\langle$*Sunny, Warm, ?, Strong, Warm, Same*$\rangle$

G_1, G_2 $\langle$?, ?, ?, ?, ?, ?$\rangle$

For training example d,

⟨Rainy, Cold, High, Strong, Warm, Change ⟩ –

S_2, $\mathbf{S_3}$ $\langle SUNNY, WARM, ?, STRONG, WARM, SAME\rangle$

$\mathbf{G_3}$ $\langle Sunny, ?, ?, ?, ?, ?\rangle$ $\langle ?, Warm, ?, ?, ?, ?\rangle$ $\langle ?, ?, ?, ?, ?, Same\rangle$

G_2 $\langle ?, ?, ?, ?, ?, ?\rangle$

For training example d,

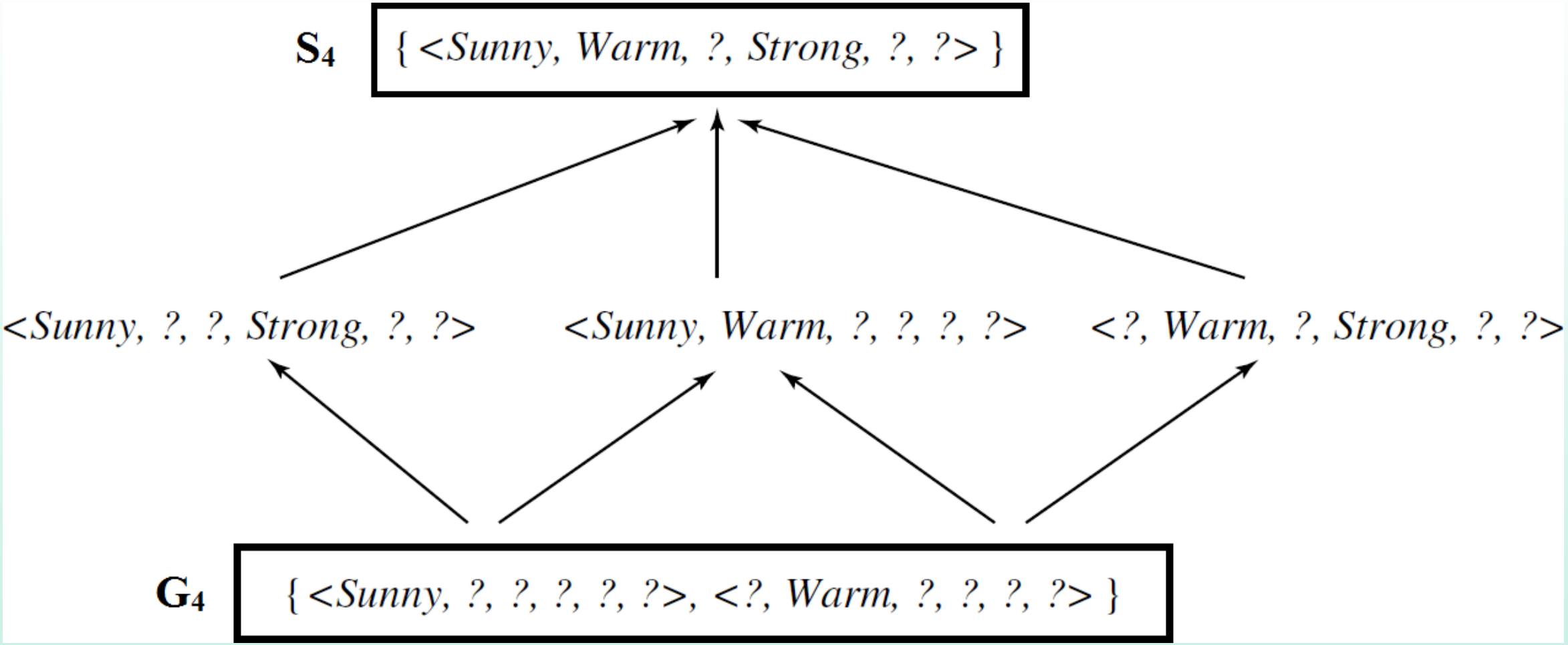
$\langle$Sunny, Warm, High, Strong, Cool Change $\rangle$ +

S_3 $\langle Sunny, Warm, ?, Strong, Warm, Same \rangle$

S_4 $\langle Sunny, Warm, ?, Strong, ?, ? \rangle$

G_4 $\langle Sunny, ?, ?, ?, ?, ? \rangle$ $\langle ?, Warm, ?, ?, ?, ? \rangle$

G_3 $\langle Sunny, ?, ?, ?, ?, ? \rangle$ $\langle ?, Warm, ?, ?, ?, ? \rangle$ $\langle ?, ?, ?, ?, ?, Same \rangle$



The final version space for the ***EnjoySport*** concept learning problem and training examples described earlier.

Convergence Of Algorithm

- Convergence Guaranteed If:
  - ***No Errors***
  - ***There Is H In H Describing C***.
- Ambiguity Removed From Vs When S = G
  - Containing Single H
  - When Have Seen Enough Tes
- If Have False Negative TE, Algorithm Will Remove Every H Consistent With TE, And Hence Will Remove Correct Target Concept From VS
  - If Observe Enough Tes Will Find That S, G Boundaries Converge To Empty VS

Example	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

LET US TRY THIS

Origin	Manufacturer	Color	Decade	Type	
Japan	Honda	Blue	1980	Economy	+
Japan	Toyota	Green	1970	Sports	-
Japan	Toyota	Blue	1990	Economy	+
USA	Chrysler	Red	1980	Economy	-
Japan	Honda	White	1980	Economy	+

AND THIS

Origin	Manufacturer	Color	Decade	Type	
Japan	Honda	Blue	1980	Economy	+
Japan	Toyota	Green	1970	Sports	-
Japan	Toyota	Blue	1990	Economy	+
USA	Chrysler	Red	1980	Economy	-
Japan	Honda	White	1980	Economy	+
Japan	Toyota	Green	1980	Economy	+
Japan	Honda	Red	1990	Economy	-

Candidate Elimination Algorithm Solved Example - 2

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 - 2

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

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: (?, ?, ?)

vtupulse.com

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

Candidate Elimination Algorithm Solved Example - 3

- S0: (0, 0, 0, 0, 0)
- S1: (0, 0, 0, 0, 0)
- S2: (Many, Big, No, Exp, Many)
- S3: (Many, ?, No, Exp, ?)
- S4: (Many, ?, No, ?, ?)

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

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

vtupulse.com

- G4: (Many,?,?,?, ?)
- G3: (Many,?,?,?, ?) (?,?,?,Exp,?)
- G2: (Many,?,?,?, ?) (?, Big,?,?,?) (?,?,?,Exp,?) (?,?,?,?,Many)
- G1: (Many,?,?,?, ?) (?, Big,?,?,?) (?,Medium,?,?,?) (?,?,?,Exp,?) (?,?,?,?,Many) (?,?,?,?,Few)
- G0: (?, ?, ?, ?, ?)