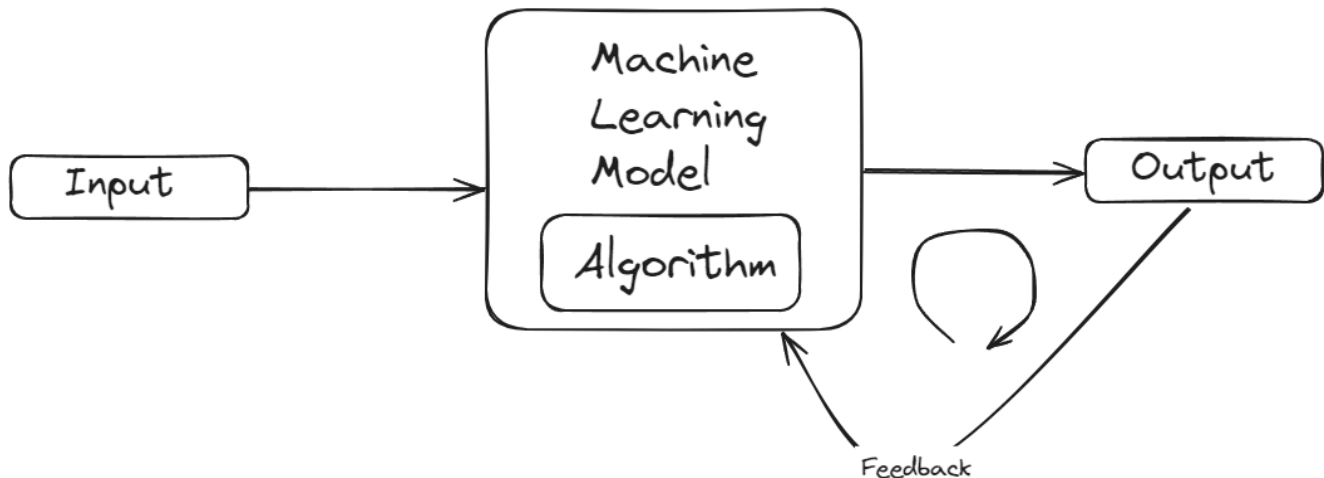


## Introduction

- Machine Learning is a branch of Artificial Intelligence (AI), which focuses on development of algorithms and statistical models that enable computers to learn and make precise decisions or predictions, about the given input data.
- These algorithms/statistical models are initially trained with the training data sets and feedback from humans.
- Hence, these learning models are completely feedback driven models.
- It follows the below steps, iteratively, to train the model and make it efficient and fully optimized.
  - + Consider a human written algorithm which takes an input, processes it, and produces an output.
    1. A machine learning model takes an input, processes it with the algorithm and produces an output.
    2. The human/automated-process gives feedback on the accuracy of the produced output, with the expected output.
    3. Based on that feedback, the model adjusts the algorithm (code) by itself without any human interventions, to meet the expected output criteria.
- Hence, ML models are feedback-driven / experience-based models which iteratively adjusts an algorithm based on the provided feedback, to meet the expected output results from the algorithm.



- Various types of ML Algorithms are:
  1. Supervised Learning: It is when the labeled data sets are provided as training data. The class labels for the new instances are predicted from that training labeled data.
  2. Unsupervised Learning: It is when unlabelled/raw data sets are provided as training data. Clustering and dimensionality reduction are two common use cases of this type of learning.
  3. Semi-Supervised Learning: It is when both labels and unlabeled data sets are provided together. Clustering and dimensionality reduction on the unlabeled data is performed based on the provided labeled data sets.
  4. Reinforcement Learning: It is when the provided feedback is in the form of rewards and penalties, based on which the model improves. Hence, it is feedback-driven and reward-based learning.

## Well Posed Learning Problems:

- An ML algorithm is built to solve a certain type of problems, such as classification, regression(prediction), clustering, anomaly detection, etc.

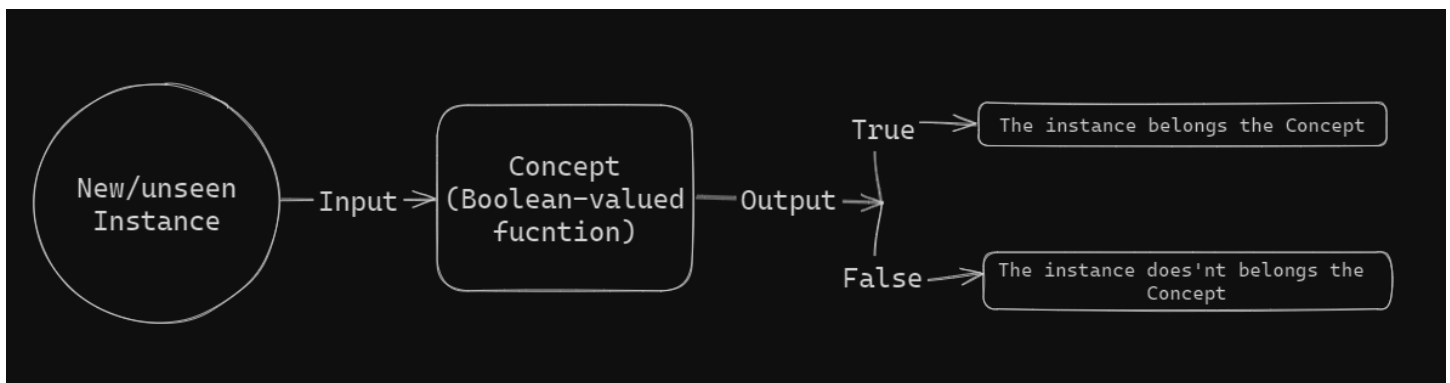
- Those problems are known as **tasks(T)** of ML models.
- An ML algorithm learns (optimizes the algorithm) based on the **experiences(E)** (Provided feedbacks/credits).
- The Performance of every task of the ML process is measured by the performance measure(P).
- Hence, An ML problem is said to be Well-posed, if it is defined with these three parameters:
  1. Task (T)
  2. Performance (P)
  3. Experience (E)
- Therefore, A computer program is said to be well-posed if it learns from the experience (E) corresponding to some tasks (T), and performance (P).
- Examples of Well-Posed Problems:
  1. Checkers:
    - Task (T): Playing Checkers
    - Performance Measure (P): Percent of games won against opponents.
    - Experience (E): Playing practice games against itself.
  2. Handwriting Recognition:
    - Task (T): Recognising and classifying handwritten words with images.
    - Performance Measure (P): Percent of words classified correctly.
    - Experience (E): A database of handwritten words with given classifications.
  3. Robot Driving:
    - Task (T): Driving on public four lane highways by using vision sensors.
    - Performance Measure (P): Average Distance traveled before error.
    - Experience (E): A Sequence of images and steering commands recorded while observing a human driver.

### Designing a Learning System:

### Perspectives and issues in Machine Learning:

### Creating learning and General to Specific Ordering:

- Concept Learning is the process of inferring (concluding based on some evidence) the general rules from the given instances (evidence).
- These inferred general rules are then used for classifying the new/unseen instances.
- Hence, Concept Learning is based on Supervised Learning.
- A Concept can be considered as a Boolean-valued function.



- Hence, Concept Learning Task is the process of learning the concept or pattern from a set of labeled data (instances with classes) and generalizing it to classify new/unseen instances.

- For Example: Consider a classification task where the goal is to distinguish b/w images of cats and dogs. (Image-classifier for cats and dogs images)
- Here, “cats” and “dogs” are the concepts.
- Features/Attributes for “cats” concept are => { pointy ears, whiskers, fur texture }
- Features/Attributes for “dogs” concept are => { floppy ears, snout shape, tail length }
- Hence the Concept Learning Task would be to generalize the provided labeled cats and dogs pictures to use the generalized function to classify the new and unclassified image of a cat / dog.

#### Notations and Terminology:

- The Generalized function is represented in the form of a Hypothesis.
- A Hypothesis(h) is a set of attributes/features which provides specific constraints for each attribute in-order to classify a new instance.
- In most of the Content Learning Tasks, the classes are always binary valued.
- Hence, an instance can be positive (true/yes) or negative(false/no).
- A new instance(x) is said to be positive if it satisfies all the constraints of a hypothesis [ i.e.,  $c(x) = 1$  ], else it is said to be negative [ i.e.,  $c(x) = 0$  ].
- Hence,

$$X \rightarrow [0, 1]$$

$c(x) = 1$  (positive)  $\Leftrightarrow$  x satisfies all the constraints specified by hypothesis h)

Else,  $c(x) = 0$  (negative)

Where,

$$h(x) = c(x) \quad \forall (x \text{ in } X)$$

$c(x)$  - boolean valued function

$X$  - set of instance

$x$  - an instance

- A Hypothesis can have (3) types of constraints:

1. “?” - any attribute value is acceptable. [ ? - placeholder for any attribute value ]
2. “ $\Phi$ ” - no value is acceptable.
3. “Attribute Value” - no other value is acceptable than the specific “Attribute value”.

- Hence,

$\langle ?, ?, ?, \dots, ? \rangle$  is known as **Most General Hypothesis**.

$\langle \Phi, \Phi, \Phi, \dots, \Phi \rangle$  is known as **Most Specific Hypothesis**.

- Specific to General Constraint Scale =  $\Phi \rightarrow \text{Attribute} \rightarrow ?$

- **Example:**

Concept: Good day for water sports.

Day	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

- Consider a hypothesis:  
 $\langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$

This Hypothesis means that “a new instance is said to be positive(yes) only if the (AirTemp = Cold) and (Humidity = High). That is, it is a good day for water sports if the air temperature is “cold” and humidity is “high”.

### **Comparing two hypothesis and determining the most general one among them:** **(General to Specific Ordering)**

Consider two hypothesis:

$h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$

$h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$

Here, Since ( $h_1$ ) has more no.of specific-constraints than that of in ( $h_2$ ), hence ( $h_2$ ) will classify more no.of instances than ( $h_1$ ). Hence ( $h_2$ ) is said to be more general than ( $h_1$ ) (or) ( $h_1$ ) is said to be more specific than ( $h_2$ ). [ i.e.,  $h_2 > h_1$  ]

### **Concept Learning as Search:**

- Concept Learning can be viewed as the task of searching through a large space of hypotheses which are defined implicitly by hypothesis representation.
- The major aim of the search is to locate the best fitting hypothesis for training examples.
- The designer of the learning algorithm defines the space for all hypotheses that the program can ever run, represent and therefore can ever learn.

Ex:

- + Consider EnjoySport Learning Task:
  - + The instance space  $X$  represents all possible combinations of attribute values for the given attributes (Sky, AirTemp, Humidity, Wind, Water, Forecast).
  - + Each attribute has a certain number of possible values: Sky (3), AirTemp (2), Humidity (2), Wind (2), Water (2), Forecast (2).
  - + The total number of distinct instances in  $X$  is calculated by multiplying the number of possible values for each attribute:  $3 \times 2 \times 2 \times 2 \times 2 \times 2 = 96$  distinct instances.
  - + The hypotheses space  $H$  represents all possible combinations of conditions or rules that can be used to classify instances.
  - + Given the attributes and their possible values, there are  $(5 \times 4 \times 4 \times 4 \times 4 \times 4 = 5120)$  syntactically distinct hypotheses within  $H$ .
  - + However, some hypotheses containing " $\emptyset$ " symbols classify every instance as negative, so they are not semantically distinct.
  - + After removing these empty hypotheses, the number of semantically distinct hypotheses reduces to  $1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3) = 973$ .
- Hence it is very crucial to study the search spaces and implement the search strategies in large or infinite hypothesis spaces, to have an effective solution and efficient process of learning.

### **Finding a maximally specific hypothesis: (Find-S Algorithm)**

[ notes ]

### **Decision Tree Learning:**

#### **Appropriate Problems for decision making:**

## **Hypothesis Space Search in Decision Tree Learning**

In decision tree learning, the **hypothesis space** is the set of all possible decision trees that could be constructed given the features and data. Searching this space involves finding the tree that best fits the training data according to some criterion. Here's a high-level overview of how this search process works:

1. **Initial Tree Construction:** The process usually starts with a root node representing the entire dataset. The algorithm recursively splits the data based on features to create child nodes, aiming to partition the data into subsets with more homogeneous labels.
2. **Splitting Criteria:** At each node, the algorithm chooses the best feature and threshold to split the data. This decision is typically based on measures such as **information gain**, **Gini impurity**, or **variance reduction**. The goal is to create child nodes that are as pure as possible, meaning they contain instances of a single class or are close to it.
3. **Recursive Search:** The algorithm continues to split the data recursively, creating branches and sub-branches, until it meets stopping criteria (e.g., maximum depth, minimum number of samples per leaf, or no further information gain). This results in a complete decision tree or a pruned version of it.
4. **Pruning:** To prevent overfitting, decision trees might be pruned. This involves removing nodes or branches that provide little predictive power, thereby simplifying the tree while retaining its generalization ability.

## Inductive Bias in Decision Tree Learning

**Inductive bias** refers to the set of assumptions a learning algorithm makes to generalize from specific training examples to unseen data. In decision tree learning, the inductive bias is influenced by several factors:

1. **Preference for Simple Trees:** Decision tree algorithms often prefer simpler trees with fewer nodes or lower depth. This bias towards simpler models helps in generalizing better and avoiding overfitting to the training data.
2. **Greedy Splitting:** The greedy approach used in decision trees (e.g., choosing the best split at each node based on criteria like information gain) is a form of inductive bias. It assumes that making locally optimal decisions will lead to a globally good tree.
3. **Axis-Parallel Decision Boundaries:** Decision trees create decision boundaries that are parallel to the axes of the feature space. This inductive bias means that decision trees might struggle with problems where optimal decision boundaries are diagonal or more complex.
4. **Categorical and Numerical Features:** Decision trees handle categorical and numerical features differently. The bias towards using categorical splits (in some algorithms) or choosing numerical thresholds for splitting can affect how well the tree generalizes.
5. **Overfitting Prevention:** Techniques like pruning and setting maximum depth are applied to balance between fitting the training data well and generalizing to new data, reflecting an inductive bias towards models that generalize better.

*[ Pruning in decision tree learning is a technique used to reduce the size of a decision tree by removing parts of the tree that provide little predictive power. The primary goals of pruning are to enhance the model's generalization ability, avoid overfitting, and improve computational efficiency. ]*