Unit-1
Introduction to ML

Prepared By: Arjun Sing Saud
Asst. Prof. CSCSIT, TU

What is Machine Learning?

- Machine learning is an application of AI that enables systems to learn and improve from experience without being explicitly programmed.
- Machine learning focuses on developing computer programs that can access data and use it to learn for themselves.
- The machine learning process begins with observations or data, such as examples, direct experience or instruction.

What is Machine Learning?

- It looks for patterns in data so it can later make inferences based on the examples provided.
- The primary aim of ML is to allow computers to learn autonomously without human intervention or assistance and adjust actions accordingly.

How Machine Learning Works?

- Learning system of a machine learning algorithm into three main parts.
- **A Decision Process**: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, your algorithm will produce an estimate about a pattern in the data.

How Machine Learning Works?

- **An Error Function**: An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
- **An Model Optimization Process**: If the model can fit better to the data points in the training set, then parameters are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluate and optimize process, until a threshold of accuracy has been met.

Applications of ML

- We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning:

1. **Image Recognition:** It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, *Automatic friend tagging suggestion*.

Applications of ML

2. **Speech Recognition:** Speech recognition is a process of converting voice instructions into text, and it is also known as *Speech to text*, or *Computer speech recognition*. Google assistant, Siri, and Alexa are using speech recognition technology to follow the voice instructions.
3. **Traffic Prediction:** If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.

Applications of ML

4. **Product Recommendations:** Machine learning is widely used by various e-commerce and entertainment companies such as *Amazon, Netflix, etc.,* for product recommendation to the user.
5. **Self-driving Cars:** Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car.

Applications of ML

6. **Email Spam and Malware Filtering:** Whenever we receive a new email, it is filtered automatically as important, normal, and spam. The technology behind this is Machine learning.
7. **Online Fraud Detection:** Machine learning is making our online transaction safe and secure by detecting fraud transaction. Outlier detection approach is primarily used in fraud detection.

Applications of ML

8. **Stock Market Trading:** Machine learning is widely used in stock market trading. Time series prediction ML models are used for the prediction of stock market trends.
9. **Medical Diagnosis:** In medical science, machine learning is used for diseases diagnoses.
10. **Machine Translation:** Machine Learning algorithms are helpful in translating text from one language to another. The technology behind the automatic translation is a sequence to sequence learning algorithm.

Machine Learning Methods

- Machine learning methods fall into three primary categories: *Supervised, Unsupervised, and Reinforcement*.

Supervised Learning:

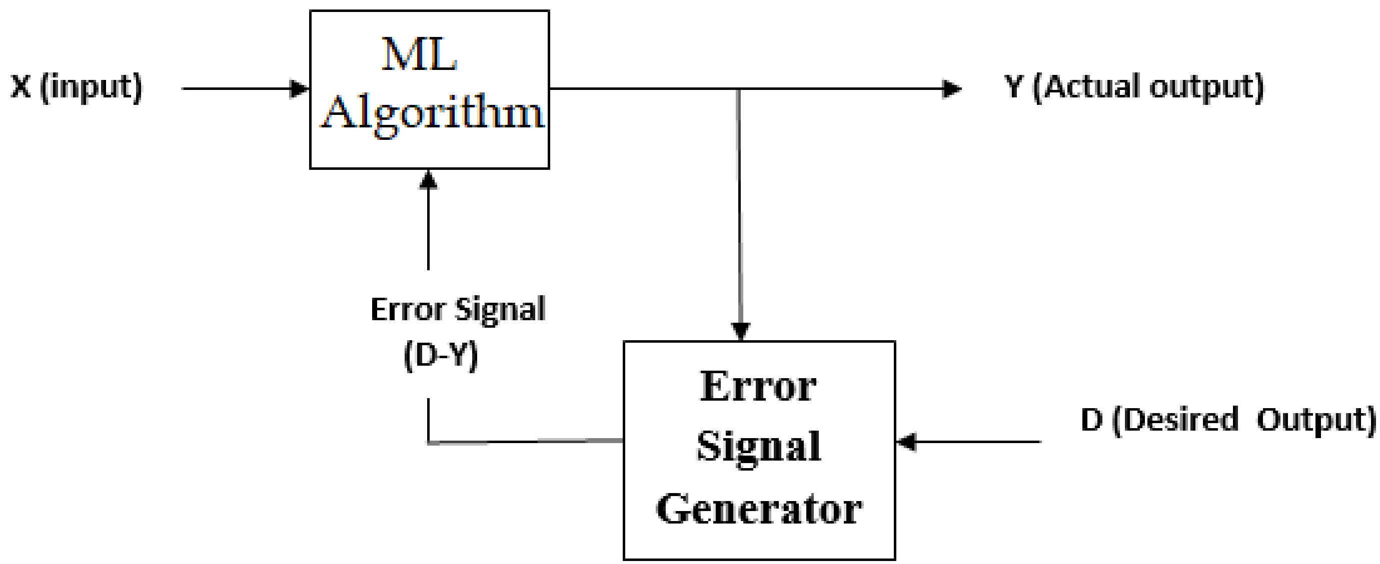
- In this learning paradigm, we present examples of correct input-output pairs to the ML algorithms during the training phase.
- This training set of examples is equivalent to the teacher for the ML algorithms.

Machine Learning Methods

- During the training of ML algorithm under supervised learning, the it takes input vector and computes output vector.
- An error signal is generated, if there is a difference between the computed output and the desired output vector.
- On the basis of this error signal, the model parameters are adjusted until the actual output is matched with the desired output.

Machine Learning Methods

- Supervised machine learning is used for performing tasks like: *Regression and Classification*.

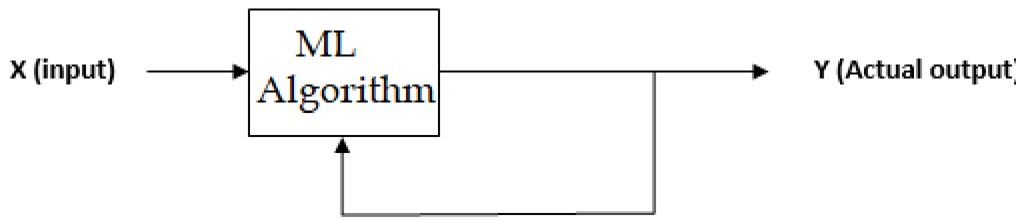


Machine Learning Methods

Unsupervised Learning

- In unsupervised learning ML algorithm is provided with dataset without desired output.
- The ML algorithm then attempts to find structure in the data by extracting useful features and analyzing its structure.
- Unsupervised learning algorithms are widely used for tasks like: *clustering, dimensionality reduction, association mining etc*.

Machine Learning Methods



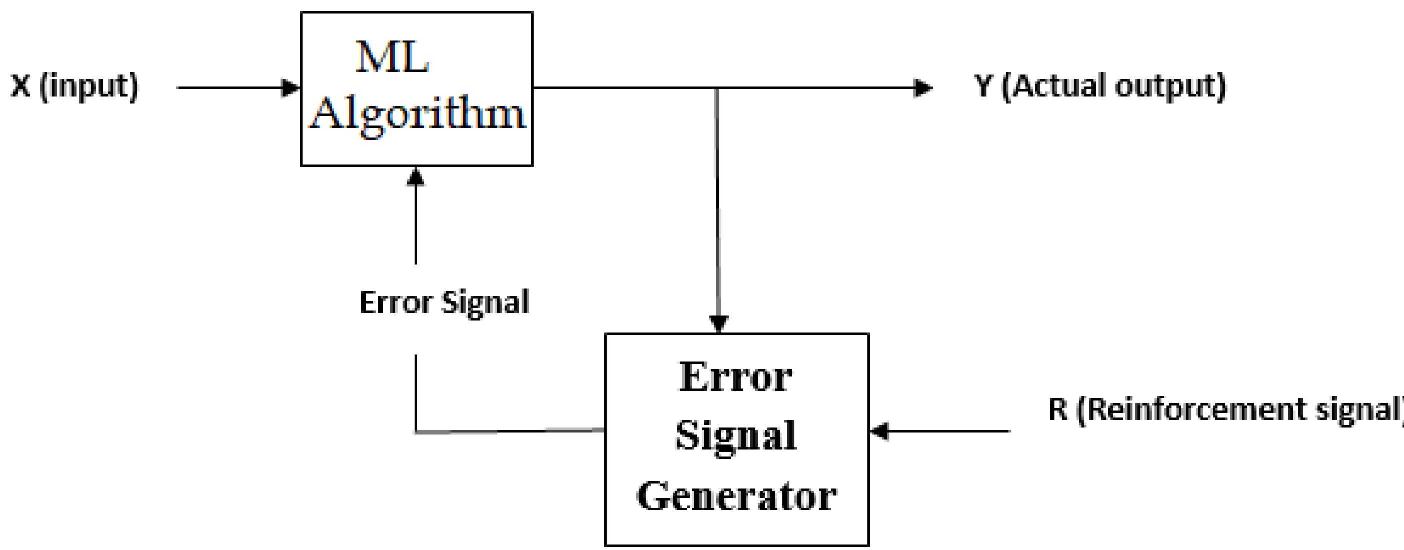
Reinforcement Learning

- In reinforcement learning, we do not provide the machine with examples of correct input-output pairs, but we do provide a method for the machine to quantify its performance in the form of a reward signal.

Machine Learning Methods

- Reinforcement learning methods resemble how humans and animals learn: the machine tries a bunch of different things and is rewarded with performance signal.
- Reinforcement learning algorithms are widely used for training agents interacting with its environment.

Machine Learning Methods



Data Normalization

- The measurement unit used can affect the data analysis. For example, changing measurement units from meters to inches for *height*, or from kilograms to pounds for *weight*, may lead to very different results.
- In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such an attribute greater effect or "weight."

Data Normalization

- To avoid dependence on the choice of measurement units, the data should be *normalized* or *standardized*. This involves transforming the data to fall within a smaller or common range such as [-1, 1] or [0, 1].
- Normalizing the data attempts to give all attributes an equal weight. There are many methods for data normalization. The major normalization methods are: *min-max normalization and z-score normalization*.

Data Transformation

- **Min-max Normalization:** It performs a linear transformation on the original data. Suppose that *min* and *max* are the minimum and maximum values of an attribute, *A*. Min-max normalization maps a value, *v* , of *A* to *nv* in the range [*new_min*,*new_max*] using following formula.

$$nv = \frac{v - \min}{\max - \min}(new\_\max - new\_\min) + new\_\min$$

Data Transformation

- **Z-score Normalization:** In z-score normalization (or zero-mean normalization), the values for an attribute, A, are normalized based on the mean and standard deviation of A. The value, v , of A is normalized to nv as below. It is also called standard normalization.

$$nv = \frac{v - \mu}{\sigma} \qquad \textit{where } \mu \textit{ is mean}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (v_i - \mu)^2}{N}},$$

where, μ is mean and n is number of data points

Data Transformation

Example

- Normalize the following data using Min-Max, and Z-score Normalization.

Salary	Age
45000	42
32000	26
58000	48
37000	32

Data Transformation

Solution

Min-Max Normalization

Salary	Age
0.50	0.73
0.00	0.00
1.00	1.00
0.19	0.27

Normalization of Salary

Salary=45000

$$nsal = \frac{45000 - 32000}{58000 - 32000}(1 - 0) + 0 = 0.5$$

Normalization of Age

Age=42

$$nage = \frac{42 - 26}{48 - 26}(1 - 0) + 0 = 0.73$$

Data Transformation

<u>Solution</u>

Z-Score Normalization

Salary	Age
0.20	0.59
-1.12	-1.29
1.53	1.29
-0.61	-0.59

Calculate mean and standard deviation of salary

$$\mu = 43000$$

$$\sigma = 9823.44$$

Now, transform salary=45000

$$nsal = \frac{45000 - 43000}{9823.44} = 0.20$$

Computational Learning Theory

- Computational Learning Theory (CoLT) is a branch of AI that focuses with formal studies on the design of computer programs that can learn.
- It is very similar to Statistical Learning Theory (SLT) as they both use Mathematical Analysis.
- The basic difference between the two is that CoLT is basically concerned with the *learnability* of the machines and the necessary steps that are required to take in order to make a given task comprehensible for an algorithm.

Computational Learning Theory

- Whereas, SLT is more focused on studying and improving the accuracy of already existing programs.
- There are many subfields of study, although perhaps two of the most widely discussed areas of study from computational learning theory are:
  - *PAC Learning*.
  - *VC Dimension*.

Computational Learning Theory

PAC Learning

- PAC Learning or Probably Approximately Correct Learning is a framework in the theory of machine learning that aims to measure the complexity of a learning problem.
- Consider that in supervised learning, we are trying to approximate an unknown underlying mapping function from inputs to outputs.

Computational Learning Theory

PAC Learning

- We don’t know what this mapping function looks like, but we suspect it exists, and we have examples of data produced by the function.
- PAC learning is concerned with how much computational effort is required to find a hypothesis (fit model) that is a close match for the unknown target function.

Computational Learning Theory

VC Dimension

- Vapnik-Chervonenkis theory, or VC theory for short, refers to a theoretical machine learning framework developed by Vladimir Vapnik and Alexey Chervonenkis.
- VC theory learning seeks to quantify the capability of a learning algorithm and might be considered the premier sub-field of statistical learning theory.
- It is a measure of the capability of a collection of functions that can be learnt by a ML algorithm in terms of complexity, expressive power, or flexibility.

Attribute Types

What is an Attribute?

- An attribute is a data field, representing a characteristic or feature of a data object. The nouns *attribute*, *dimension*, *feature*, and *variable* are often used interchangeably in the literature.
- Machine learning literature tends to use the term *feature*, while statisticians prefer the term *variable*. Data mining and database professionals commonly use the term *attribute*.
- Attributes describing a customer object can include, for example, *customer ID*, *name*, and *address*.

Data Objects and Attribute Types

Types of Attributes

On the basis of set of possible values attributes can be divided into following types

- Nominal Attributes
- Ordinal Attributes
- Interval-scaled Attributes
- Ratio-scaled Attributes

Data Objects and Attribute Types

Nominal Attributes

- The values of a **nominal attribute** are symbols or *names of things*.
- Each value represents some kind of category, code, or state, and so nominal attributes are also referred to as **categorical**. The values do not have any meaningful order.
- Examples of nominal attributes:
  - ✓Hair_color: possible values are: {black, brown, red, grey, white}

Data Objects and Attribute Types

Nominal Attributes

- ✓ Marital_status: possible values are:{Married, Single, Divorced, Widowed}
- ✓ Customer_ID: possible values are: Combination of numbers

- It is possible to represent such symbols with numbers. With *hair_color*, for instance, we can assign a code of 0 for *black*, 1 for *brown*, and so on. However, in such cases, the numbers are not intended to be used quantitatively.

Data Objects and Attribute Types

Ordinal Attributes

- An **ordinal attribute** is an attribute with possible values that have a meaningful order or *ranking* among them, but the magnitude between successive values is not known.
- Examples of ordinal attributes:
  - ✓Grades: possible values are: {A+, A, A-, B+, B, B- and so on}
  - ✓Height: possible values are:{Tall, Medium, Short}

Data Objects and Attribute Types

Ordinal Attributes

- The values have a meaningful sequence (which corresponds to increasing height); however, we cannot tell from the values *how much* bigger, say, a medium is than a short.
- Note that nominal, and ordinal attributes are *qualitative*. That is, they *describe* a feature of an object without giving an actual size or quantity.
- We can compute median and mode of ordinal attributes. However, we cannot compute mean.
- But, we can only compute mode of nominal attributes.

Data Objects and Attribute Types

<u>Interval-Scaled Attributes</u>

- Interval-scaled attributes are numeric attributes. A numeric attribute is *quantitative*; that is, it is a measurable quantity, represented in integer or real values.
- The values of interval-scaled attributes have order and can be positive, 0, or negative. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the *difference* between values.
- Because interval-scaled attributes are numeric, we can compute their mean value, in addition to the median and mode measures of central tendency.

Data Objects and Attribute Types

Interval-Scaled Attributes

- A *temperature* attribute is interval-scaled. Suppose that we have the outdoor *temperature* value for a number of different days, where each day is an object. By ordering the values, we obtain a ranking of the objects with respect to *temperature*.
- In addition, we can quantify the difference between values. For example, a temperature of 20^0C is five degrees higher than a temperature of 15^0C.
- Calendar dates are another example. For instance, the years 2002 and 2010 are eight years apart.

Data Objects and Attribute Types

Interval-Scaled Attributes

- Temperatures in Celsius and Fahrenheit do not have a true zero-point, that is, neither 0^0C nor 0^0F indicates "no temperature."
- Although we can compute the *difference* between temperature values, we cannot talk of one temperature value as being a *multiple* of another.
- Without a true zero, we cannot say, for instance, that 10^0C is twice as warm as 5^0C. That is, we cannot speak of the values in terms of ratios. Similarly, there is no true zero-point for calendar dates.

Data Objects and Attribute Types

Ratio-Scaled Attributes

- A **ratio-scaled attribute** is a numeric attribute with an inherent zero-point.
- That is, if a measurement is ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value.
- In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.
- Temperature in Kelvin, length, counts, elapsed time, etc. are examples of ratio scaled attributes