Date / / /

UNIT-1 : INTRODUCTION

- Well-posed learning problems
- Designing a learning system
- Perspectives and Issues in Machine Learning
- Examples of ML Applications
- Learning Associations
- classification
- Regression
- Unsupervised learning
- Reinforcement learning
- Supervised learning
- Concept learning
- General-to-specific ordering
- A Concept-learning task
- Concept learning as search
- Find-S

Date: / / /

What is Machine Learning?

Machine Learning is a "field of study that gives computers the capability to learn without being explicitly programmed."

ML can be explained as automating and improving the learning process of computers based on their experience without being actually programmed. i.e., without any human assistance

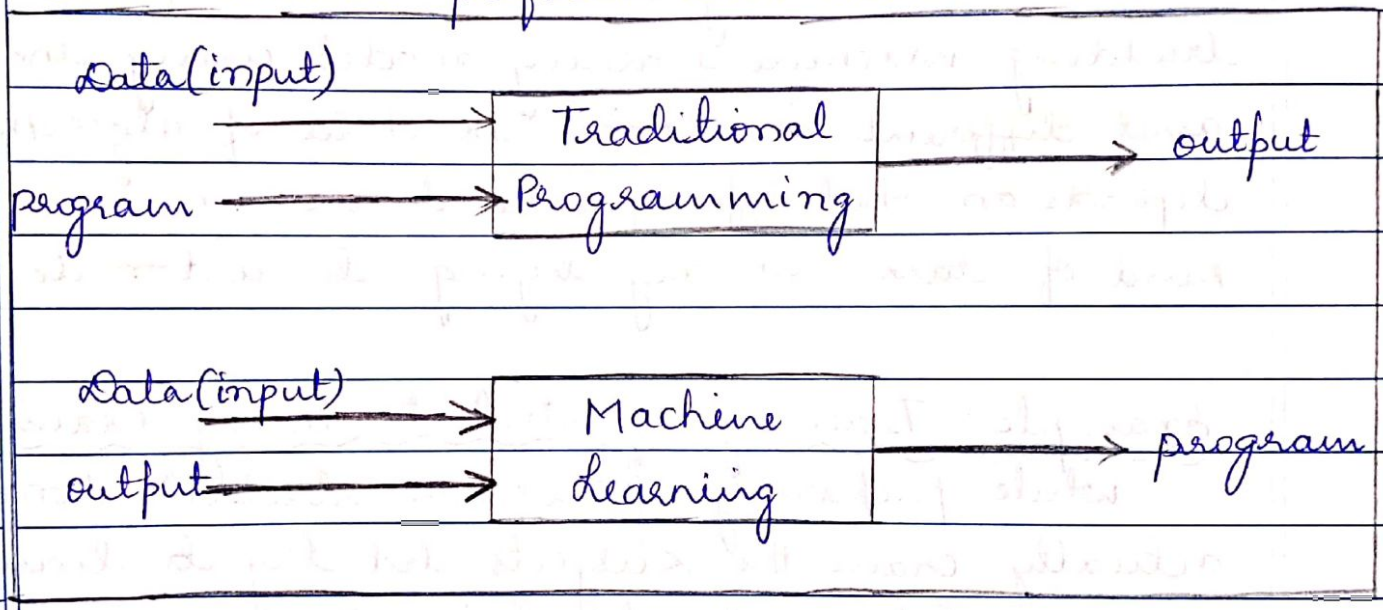
The process starts with feeding good quality data and then training our machines (computers) by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data do we have and what kind of task we are trying to automate.

Example: Training of students during exam.

While preparing for exams, students don't actually cram the subjects but try to learn it with complete understanding. Before the examination, they feed their machine (brain) with a good amount of high-quality data (questions and answers from different books/teachers notes/online Video lectures). Actually they are training their brain with input as well as output i.e. what kind of approach or logic do they have to solve a different kind of questions.

Each time they solve practice test papers and find the performance (accuracy/score) by comparing answers with answer key given. Gradually, the

performance keeps on increasing, gaining more confidence with the adopted approach. That's how actually models are built, train machine with data (both inputs and outputs are given to model) and when the time comes test on data (with input only) and achieves our model scores by comparing its answer with the actual output which has not been fed while training. Researchers are working with assiduous efforts to improve algorithms, techniques so that these models perform even much better.



Basic differences between ML and Traditional programming.

- Traditional Programming
  we feed in Data(input) + Program(logic), run it on machine and get output

- ML.
  we feed in Data(input) + output, run it on machine during training and the machine

creates its own program (logic), which can be evaluated while testing.

Well-posed learning problems.

A Computer is said to be learning from Experiences E with respect to some class of Tasks T and Performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Example 1: playing checkers

E = the experience of playing many games of checkers

T = the task of playing checkers.

P = the probability that the program will win the next game.

Example 2: Handwriting recognition

E = A database of handwritten words with given classification

T = recognizing and classifying handwritten words within images

P = Percent of words correctly classified.

Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised Machine Learning Algorithms

These algorithms can apply what has been learned

Date: / /

in the past to new data using labeled examples to predict future events.

Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct intended output and find errors in order to modify the model accordingly.

Unsupervised Machine Learning Algorithms

These are used when the information used to train is neither classified or (trained) labeled Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system can't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Semi-supervised Machine Learning Algorithms

It falls somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training - typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning

Date: / / /

accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it/ learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

Reinforcement Machine learning Algorithms

It is a learning method that interacts with its environment by producing actions and discovers errors & rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

(AI) (ML) Date: / / /

Artificial Intelligence Vs Machine Learning Vs Deep Learning (DL)

Machine Learning

Data Mining is a technique of examining a large pre-existing database and extracting new information from that base database, its easy to understand, right, ML does the same, infact, ML is a type of DM technique.

Basic definition of ML

"ML is a technique of parsing data, learn from that data and then apply what they have learned to make an informed decision."

Now a days many of big companies use ML to give their users a better experience.

Eg: * Amazon using ML to give better product choice recommendations to their costumers based on their preferences.

* Netflix uses ML to give better suggestions to their users of the TV series a movie or shows that they would like to watch

Deep Learning

DL is a subset of ML. It is technically is ML and functions in the same way but it has different capabilities.

Date: / / /

The main difference between deep and ML is, ML models become better progressively but the model still needs some guidance. If a ML model returns an inaccurate prediction then the programmer needs to fix that problem explicitly but in the case of DL, the model does it by himself. Automatic car driving system is a good example of DL.

An example to understand DL and ML.
Suppose we have a flashlight and we teach a ML model that whenever someone say "dark" the flashlight should be on, now the ML model will analyse different phrases said by people and it will search for the word "dark" and as the words comes the flashlight will be on but what if someone said "I am not able to see anything the light is very dim", here the user wants the flashlight to be on but the sentence doesnot consist the word "dark" so that flashlight will not be on. That's where DL is different from ML. If it were a DL model it would on the flashlight, a DL model is able to learn from its own method of computing.

Artificial Intelligence:

It is completely a different thing from ML and DL. DL and ML are the subsets of AI.

"AI is a ability of computer program to function like a human brain".

AI means to actually replicate a human brain, the way a human brain thinks, works and functions.

Eg:- Sophia, the most advanced AI model present today.

ML, DL and AI

ML and DL is a way of achieving AI., which means by the use of ML and DL we may able to achieve AI in future but it is not AI.

Date: / / /

Designing a learning System

Issues / Approaches in Design

* Choosing the training experience
* Choosing the target function
* Choosing the representation for the target function
* Choosing a function approximation algorithm
  - Estimating training values
  - Adjusting the weights
* The final design

Issue 1: Choosing the training experience

The type of training experience available can have a significant impact on success or failure of the learner

The key attributes are,

(1) Direct or Indirect feedback provided by the training experience

Eg: In learning to play Checkers game.

Direct feedback: For every board state, the correct move is stated

Indirect feedback:

The move sequences and the final outcome (result) of the game is known & the system has to analyse the moves depending on the result. (i.e., the correctness of the moves must be inferred from the outcome of the game)

Here, the learner faces an additional problem of credit assignment, or determining the degree to which each move in the sequence deserves credit or blame for the final outcome.

NOTE: Direct feedback is better than indirect feedback (for learning).

(2) Degree to which the learner controls the sequence of training examples
- The learner might rely on the teacher to select informative board states & to provide the correct move for each
- The learner might itself propose the board states that it finds confusing & ask the teacher for the correct move
- The learner may have control over both the board states & training classifications - as it does when it learns by playing against itself with no teacher present. (The learner might come up with novel board states and experiments).

(3) How well the training experience represents the distributions of examples over which the final system performance P must be measured
- The training examples must follow a distribution similar to that of

Date: / / /

future test examples.

Eg:- In checkers board, if the training is only games played against itself, there is danger, as this experience is not enough to complete against a human expert.

Issue 2: Choosing the target function

- Determine exactly what type of knowledge will be learned and how this will be used by the performance program.

Eg:- Checkers - playing program : For a board state, there might be many legal moves, the program needs to know the best move from these legal moves. Here multiple legal moves represent a large search space & a best move needs to be selected. Many optimization problems fall into this class.

Lets denote this as a function - (function that chooses the best move for any given board state)

$$\text{ChooseMove} : B \rightarrow M$$

ChooseMove → function that accepts as input any board from the set of legal board states B and produces as output some move from the set of legal moves (M)

In machine learning, the problem of improving the performance P at task T can be reduced to some particular target function (Eg:- chooseMove)

ChooseMove turns out to be very difficult to learn given kind of indirect learning experience. The easier alternative function would be an evaluation function that assigns a numerical score to any given board state

$V : B \rightarrow R$

(target function) (legal board state) (Real Number)

V assigns higher scores to better board states.

Issue 3: choosing a representation for the target function

The approximation of the target function V given by $\hat{V}$ must be represented in some form.

The different forms should/could be large table with a distinct entry specifying the value for each distinct board state, or a quadratic polynomial function of predefined board features or an artificial neural network.

The more expressive the representation, the more training data the program will require inorder to choose among the alternative hypotheses it can represent.

To make it simple, $\hat{V}$ will be calculated as a linear combination of the following board features

x_1 : the number of black pieces on the board
x_2 : the number of red pieces on the board
x_3 : the number of black kings on the board
x_4 : the number of red kings on the board
x_5 : the number of black pieces threatened by red
x_6 : the number of red pieces threatened by black.

Thus the learning program will represent $\hat{V}(b)$ as a linear function of the form

$$\hat{V}(b) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6$$

where, w_0 through w_6 are numerical coefficients, or weights to be chosen by the learning algorithm.

Learned values for the weights w_1 through w_6 will determine the relative importance of the various board features in determining the value of the board.

The net effect of design choices is to reduce the problem of learning a checkers strategy to the problem of learning values for the coefficients w_0 through w_6 in the target function representation

Issue 4: Choosing a function Approximation Algorithm

In order to learn a target function $\hat{V}$, we require a set of training examples, each describing a specific board state b & the training value $V_{train}(b)$ for b.

The training pair is an ordered pair of the form $<b, V_{train}(b)>$

Eg:- board state b in which black has won the game (note $x_2 = 0$ indicates that red has no remaining pieces and for which target value $V_{train}(b)$ is therefore +100

$$<<x_1 = 3, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 0>, +100>$$

The Final Design

The final design of our checkers learning system can be naturally described by four distinct program modules that represent the central components of many learning systems.

- The performance System is the module that must solve the given performance task, in this case playing checkers, by using the learned target function(s). It takes an instance of a new problem (new game) as input and produce a trace of its solution (game history) as output.

In our case, the strategy used by the performance system to select its next move at

Date: / / /

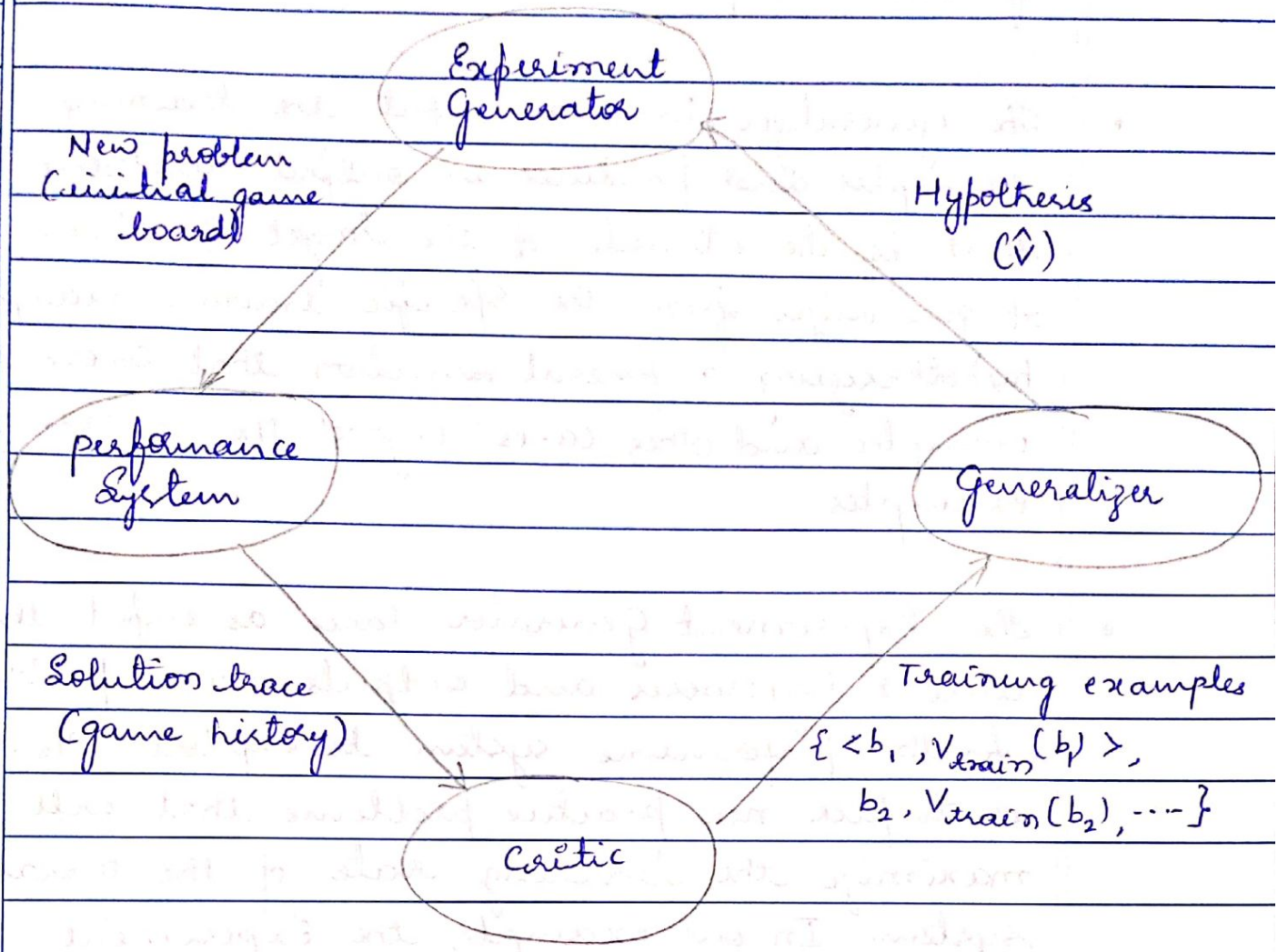


fig: Final design of the checkers learning program

each step is determined by the learned $\hat{V}$ evaluation function. Therefore, we expect its performance to improve as this evaluation function becomes increasingly accurate.

- The Critic takes as input the history or trace of the game and produces as output a set of training examples of the target function. As shown in the diagram, each training example in this case corresponds to some game state in the trace, along with an estimate V_{train} of the target function value

Date: / /

for this example

- The Generalizer takes as input the training examples and produces an output hypothesis that is its estimate of the target function. It generalizes from the specific training examples, hypothesizing a general function that covers these examples and other cases beyond the training examples.

- The Experiment Generator takes as input the current hypotheses and outputs a new problem for the performance system to explore. Its role is to pick new practice problems that will maximize the learning rate of the overall system. In our example, the Experiment Generator follows a very simple strategy. It always proposes the same initial game board to begin a new game

Date: / /

Issues in Machine Learning

- 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? what algorithms perform best for which types of problems & representations?

- How much training data is sufficient? what general bounds can be found to relate the confidence is learned hypothesis to the amount of training experience and the character to the learner's hypothesis space?

- 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?

- what is the best strategy for choosing a useful next training experience, and how does the choice of this strategy after the complexity of the learning problem?

- 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?

Date: / / /

- How can the learner automatically alter its representation to improve its ability to represent and learn the target function?

Concept learning and the General-to-Specific ordering

what is learning?

There are several definitions of "learning". One of the simplest definitions is,

"The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something."

Inferring a boolean-valued function from training examples of its input and output

Concept

can be viewed as describing some subset of objects or events defined over a large set

Concept learning task

Each hypothesis is a conjunction of constraints on the instance attributes

Let each hypothesis be a vector of six constraints specifying the value of six attributes : Sky, AirTemp,

Humidity, Wind, Water, Forecast.

Example	Sky	AirTemp	Humidity	Wind	water	forcast	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

Table: Positive and negative training examples for the target concept EnjoySport.

For each attribute, the hypothesis will either be

- indicate by a "?" that any value is acceptable for this attribute
- Specify a single required value (Eg: Warm) for the attributes or
- Indicate by a "ϕ" that no value is acceptable.

If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example ($h(x) = 1$),

The most general hypothesis - that everyday is a positive example - is represented by.

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

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

$\langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$

Date: / / /

Notation:

The set of items over which the concept is defined is called the set of instances, which we denote by X.

The concept or function to be learned is called the target concept, which is denoted by c.

In general, c can be any boolean-valued function defined over the instances X; that is,

$c : X \rightarrow \{0, 1\}$.

Eg:- the target concept corresponds to the value of the attribute EnjoySport

(c(x) = 1 if EnjoySport = Yes, and
c(x) = 0 if EnjoySport = No)

Given a set of training examples of the target concept c, the problem faced by the learner is to hypothesize, or estimate, c.

We use the symbol H to denote the set of all possible hypotheses that the learner may consider regarding the identity of the target concept. Usually H is determined by the human designer's choice of hypothesis representation.

In general, each hypothesis h in H represents a boolean-valued function defined over X; that is,

$h : X \rightarrow \{0, 1\}$.

The goal of the learner is to find a hypothesis h such that $h(x) = c(x)$ for all x in X.

Date: / /

General-to-Specific ordering of Hypothesis

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

$h_1 = \langle Sunny, ?, ?, Strong, ?, ? \rangle$

$h_2 = \langle Sunny, ?, ?, ?, ?, ? \rangle$

Now, consider the set of instances that are classified positive by h_1 and by h_2. Because h_2 imposes fewer constraints on the instance, it classifies more instances as positive. In fact, any instances classified positive by h_1 will also be classified positive by h_2. Therefore, we say h_2 is more general than h_1.

"More general than" relationship between hypothesis can be defined more precisely as follows.

Definition: Let h_j and h_k be boolean valued function 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

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

Date / / /

Find-S: Finding a Maximally Specific Hypothesis

Find-S Algorithm

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

1. The first step is, find the most specific h

   $h \leftarrow \langle \phi, \phi, \phi, \phi, \phi, \phi \rangle$

   meaning, EnjoySport is a no for all possible instances

2. Look at the first +Ve example & modify h

   $h \leftarrow \langle$ Sunny, warm, Normal, Strong, warm, same$\rangle$

3. Look at the 2nd +Ve example & modify h accordingly

   $h \leftarrow \langle$ Sunny, warm, ?, Strong, warm, same$\rangle$

4. Look at the next +Ve example (4th instance) & modify h

   $h \leftarrow \langle$ Sunny, warm, ?, Strong, ?, ?, $\rangle$

   This is the hypothesis

Issues of find-S

1) Has the learner converged to the correct target concept?
2) why prefer the most specific hypothesis? Especially when there are multiple hypotheses consistent with the training examples.
3) Are training examples consistent? what about noise / error
4) what if there are several maximally specific consistent hypotheses (should we back tract?)

Dimensionality Reduction

what?

The process of selecting a subset of features for use in model construction.

Pre-processing of data in Machine Learning

why?

In the observed data (input data) there is abundance of redundant & irrelevant features.

- If features are more, then model becomes more complex. i.e., complexity increases with dimensionality
- Increase in complexity, increases Computational time, number of training samples needed & also complexity of the inference algorithms.

Date: / / /

- Simpler models are more robust on small datasets.
- It is easy for plotting & visualizations for structure & outliers.
- execution time is reduced
- Simpler explanations are generally better than complex ones
- Also, search space would be larger & searching in large space becomes difficult.

How ?

1) Feature Selection ← (wrapper Method)
2) Feature Extraction ← (Filter Method)

Feature Selection

This involves finding 'k' of the 'd' dimensions that gives us the most information, & we discard the other (d-k) dimensions.

Feature Extraction

Involves finding a new set of 'k' dimensions that are combinations of the original 'd' dimensions

Feature Selection.

- → Supervised Methods (wrapper)
- → Unsupervised Methods (filters)

Subset selection : Ex supervised feature selection.

It involves finding the best subset of the set of

features. The best subset contains the least number of features (dimensions) that most contribute to accuracy. The remaining unimportant dimensions can be discarded.

Factor Analysis: (FA)

FA identifies correlations between and among variables to bind them into one underlying factor driving their values

Correlational Analysis

→ Correlation → A way of representing the relationship between two variables

→ Bivariate → Refers to two variables

→ Correlational Analysis often referred to as Bivariate Correlational Analysis.

→ Correlations are viewed as strong or weak.

A sister or brother — strong relation

4th or 5th Cousin — weak relation.

→ Correlations could be +ve or -ve

A friend or A enemy