Introduction

* Popularity of machine learning field in recent time and the reasons behind that
  - New software / algorithms
    * Neural networks
    * Deep learning — grew out of work in AI
  - New hardware — New capability for computers
    * GPU's
  - cloud Enabled
  - Availability of Big Data

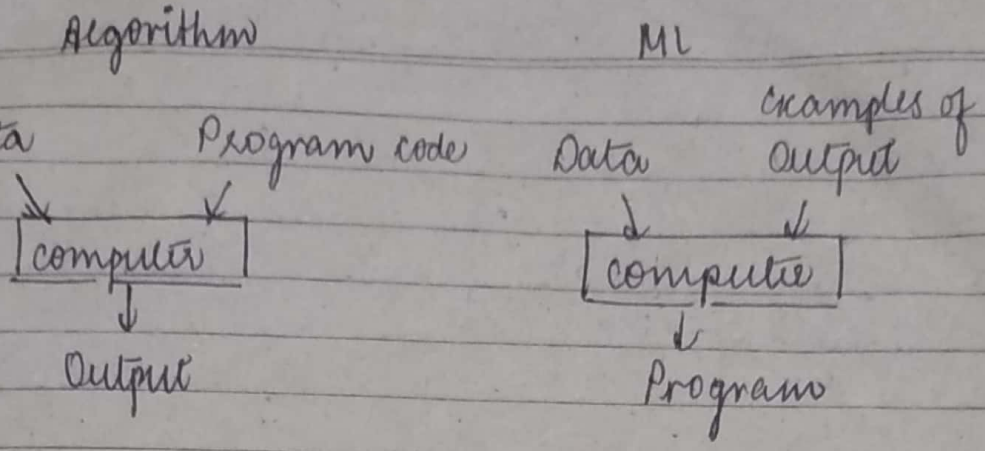
2009 - Google builds self driving car.

2011 - Watson wins jeopardy.

2014 - Human vision surpassed by ML systems.

How a machine learning soln differs from a programatic soln -

If given an algorithm / program, the computer program takes data as input and produces output. On the other hand, when we use machine learning, the data input as well as examples of output are feeded to the computer and we get a program / model with which we can solve subsequent tasks.



Learning - the ability to improve once behavior based on experience.

machine learning explores algorithms that learn / build models for data that can be used for different tasks such as prediction, decision making or solving tasks.

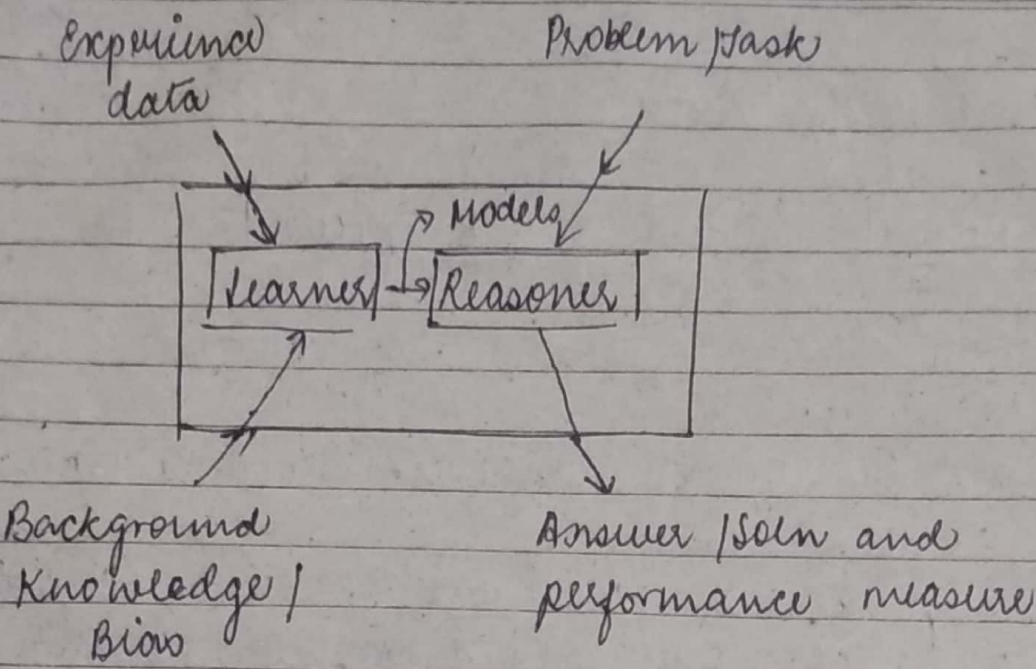
formal definition of machine learning →

Tom Mitchell definition for ML :-
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P,
if its performance on Tasks in T, as measured by P, improves with experience E.

Where
task T → example → prediction, classification, acting in an environment.
(behaviours)
Experience E → data

Based on this definition, we can see our learning system as -

Schematic diagram of a ML system



The learner takes experience & background knowledge and builds models. These models can be used by the reasoner which given a task finds soln to the task and corresponding performance measures.

Many domains and applications –

* Medicine :-

⇒ diagnose a disease

– Input :- symptoms, lab measurements, test results, DNA tests, ...

– Output :- one of set of possible diseases, or "none of the above".

⇒ Data mine historical medical records to learn which future patients will respond best to which treatments.

* Vision :-
⇒ say what objects appear in an image.
⇒ convert hand - written digits to characters 0...9.
⇒ detect where objects appear in an image.

* Robot control :-
design autonomous mobile robots that learn to navigate from their own experience.

* NLP (national language Processing) :-
⇒ detect where entities are mentioned in NL.
⇒ detect what facts are expressed in NL.
⇒ detect if a product / movie review is positive, negative, or neutral.

* Speech recognition
* Machine Translation

* Financial :-
⇒ predict if a stock will rise or fall
— in the next few milliseconds.
⇒ predict if a user will click on an ad or not
— in order to decide which ad to show.

Application in Business Intelligence

* Robustly forecasting product sales quantities taking ~~seo~~ seasonality and trend into account.

* Identifying cross selling promotional opportunities for consumer goods.

* Identify the price sensitivity of a consumer product and identify the optimum price point that maximizes net profit.

* Optimizing product location at a super market retail outlet.

* Modelling variables impacting customers churn and refining strategy.

* Fraud detection :- Credit card providers

* determine whether or not someone will default on a home mortage.

* understand consumer sentiment based off of unstructured text data.

* Forecasting women's conviction rates based off external macroeconomic factors.

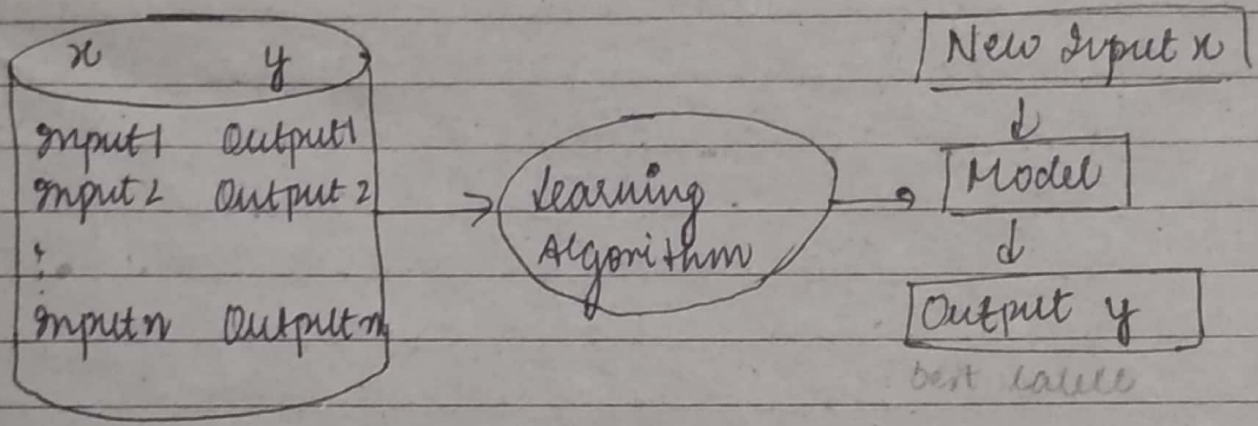
Training experience = features → more difficult to learn
represent the target func. = Rich representation = class of func
How to create a learner :- = hypothesis lang.

1) choose the training experience.
2) choose the target function (that is to be learned)
3) choose how to represent the target function.
4) choose a learning algorithm to infer the target function.

Different types of learning

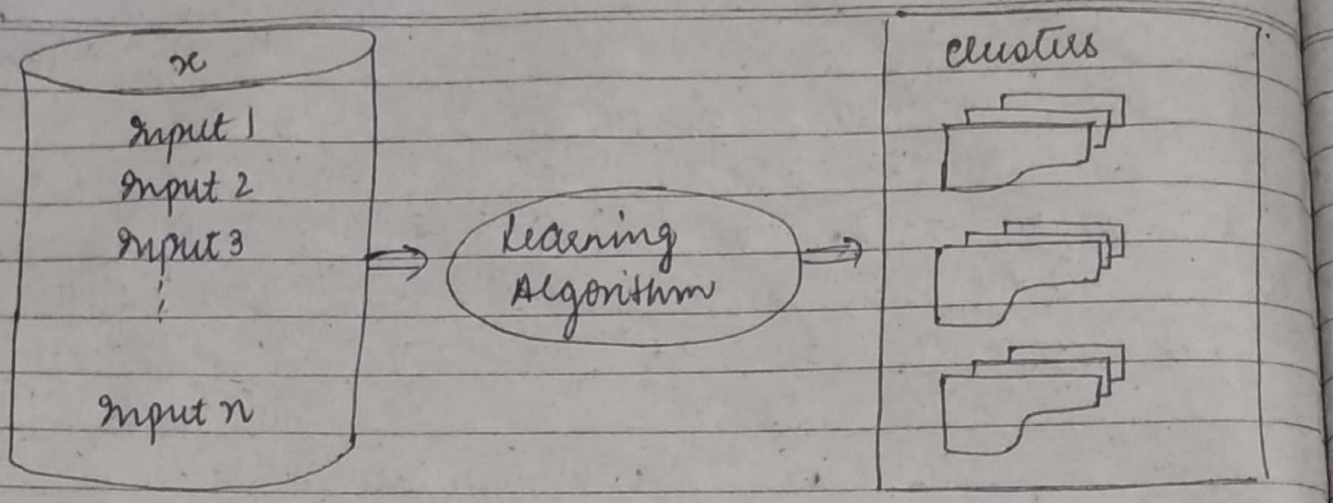
1) Supervised learning :-

If we take pre-classified training examples where x is an input and y is an output. We find the best label for y at a given observation x at every instance. Thus, it is called labelled data.



2) Unsupervised learning :-

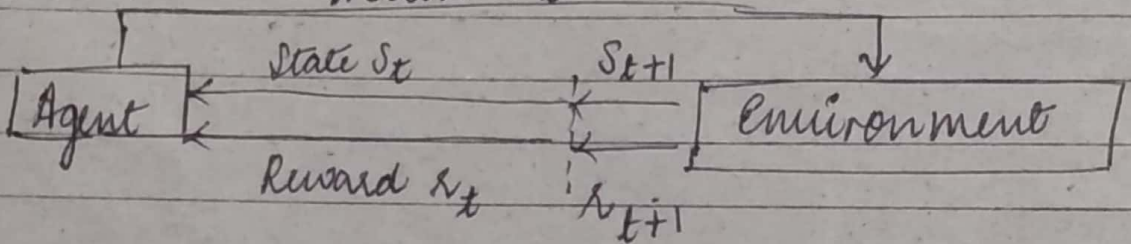
Here we have no labels, only for a given set of x's we have to summerize or cluster the data (points).



3) Reinforcement Learning -

An agent acts in an environment and we figure out what actions the agent must take at every step.

The agents determine the actions based on rewards & punishments.



4) Semi- supervised learning :-

It is a combination of supervised and unsupervised learning. This means we have some labelled data and larger amount of unlabelled data so we have to come up with some learning algo mat can convert even when the training data is limited.

Some more examples of ML are -

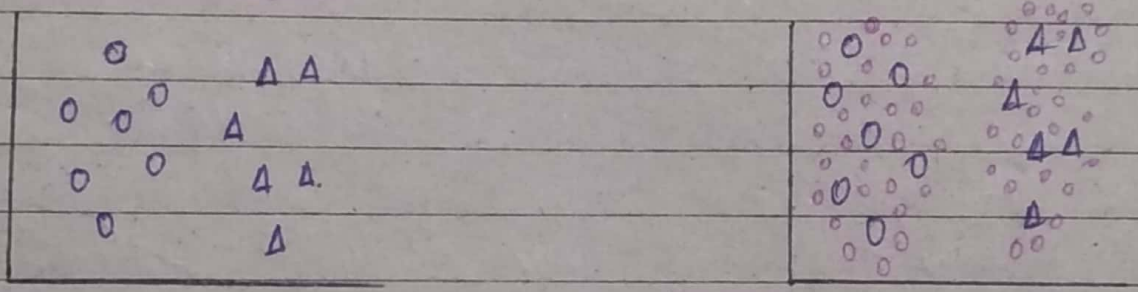
- database mining
large datasets from growth of automation / web.
eg - web click data, medical records, biology, engineering.

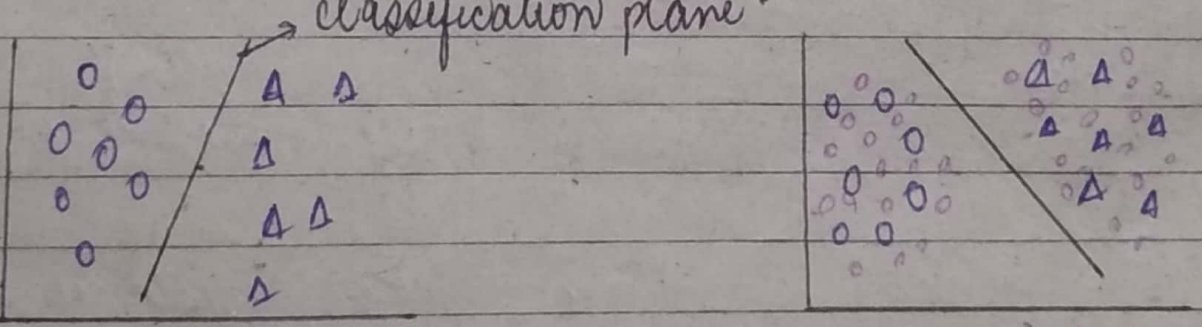
- Applications can't program by hand
eg - autonomous helicopter, handwriting recognition, most of Natural Language Processing (NLP), Computer Vision.

- Self - customizing programs.
Eg; Amazon, Netflix product recommendations.

- Understanding human learning (brain, real AI).

Semi supervised learning.





Supervised learning

Here we have a data set that includes target values (the values we wish to predict). We try to learn a function that correctly predict the target values from the other features, which can then be used to make predictions about other examples.
Typical examples - classification, regression.

Given :
- a set of input features $x_1, \ldots, x_n$ to describe instances
- A target feature Y
- a set of training examples where the values for the input features and ~~output~~ the target features are given for each example
- a new example, where only the values for the input features are given.

Predict the values for the target features for the new example.
- classification when Y is discrete
- regression when Y is continuous.

	X_1	X_2	.....	X_n	Y	
I_1	a_1	a_2	.....	a_n	y_1	] Instance / Examples
I_2	b_1	b_2	.....	b_n	y_2	} training instance
I_3	c_1	c_2	.....	c_n	y_3	} training instance
	z_1	z_2	.....	z_n	?	Testing instance

Y (target feature) can be discrete or continuous valued.

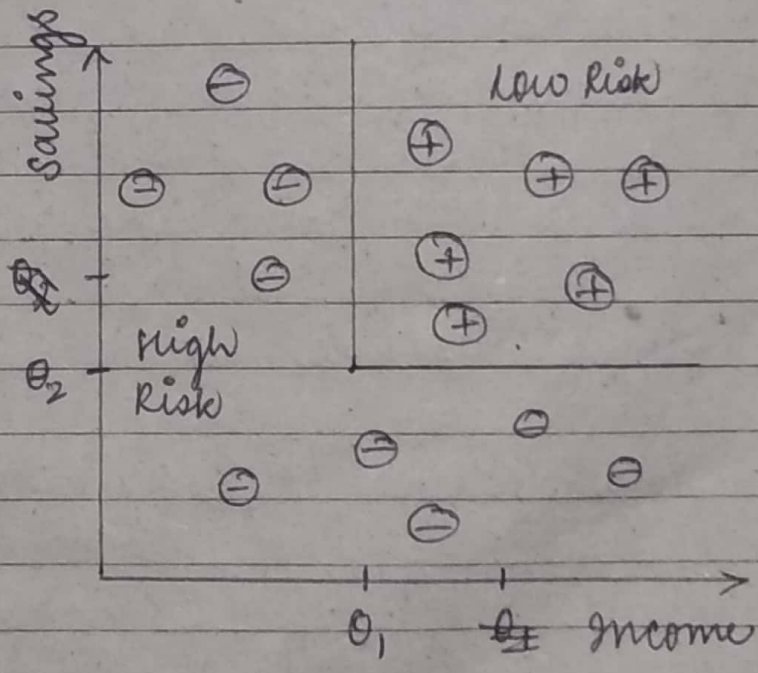
- discrete valued are classification problems such as whether it will rain tomorrow or not / given symptoms of the patient we can predict the patient has particular disease.

- continuous valued are regression problems such as given values of features (location, floor area, no. of rooms etc) we want to predict the price of a house.

Examples of classification problems -

1) Credit Scoring :- Differentiating between low-risk and high risk customers from their income and savings.

Discriminant :-
(the comeup result of this)
If income $> \theta_1$ AND
savings $> \theta_2$ then
low risk
else
High risk.



Example of Regression

Here we come up with functions with takes (input and parameters) as instance.

Example -

Price of a used car

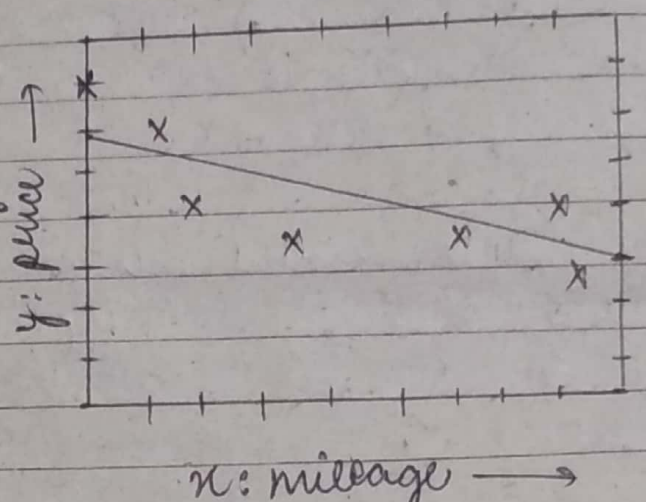
x : car attributes

y : price

$y = g(x, \theta)$

$g()$ → model &

θ → parameters

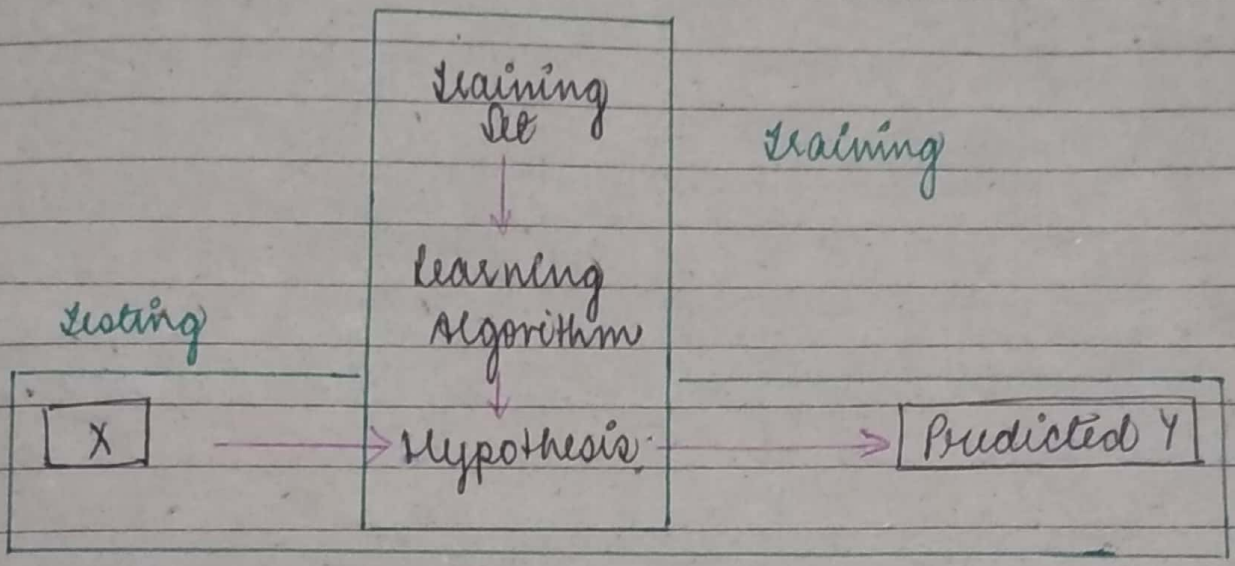


Features :-

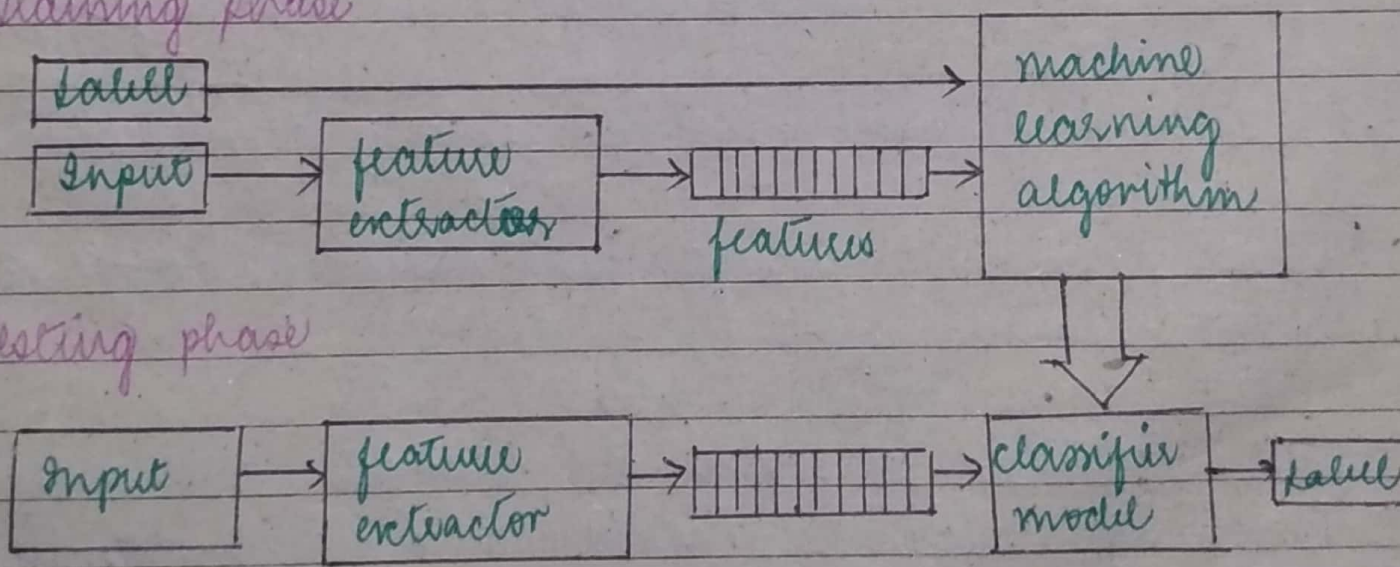
Often, the individual observations are analyzed into a set of quantified properties which are called features. May be

- categorical (eg. "A", "B", "AB" or "O", for blood type) ("male", "female", for gender type)
- ordinal (eg. "large", "medium" or "small")
- integer-valued (eg. the number of words in a text)
- real-valued (e.g. height)

the schematic diagram for ~~schematic~~ supervised learning



In supervised learning, the learning algorithm uses the training set to come up with a model/ hypothesis. Then in the testing phase, given a new instance we use the hypothesis to predict the value of y.



Hypothesis Space :-

One way to think about a supervised learning machine is as a device that explores a "hypothesis space".

- Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

Terminologies

Features :- The no. of features or distinct traits that can be used to describe each item in a quantitative manner.

Feature vector :- n - dimensional vector of numerical features that represents some object (multiple features are described through feature vector).

Instance Space X :- Set of all possible objects described by features.

Example (x, y) :- Instance x with label $y = f(x)$

Concept c :- Subset of objects from X (c is unknown)

Target function f :- Maps each instance $x \in X$ to target label $y \in Y$.

Training Data :- Collection of examples observed by learning algorithm : Used to discover potentially predictive relationships.

Hypothesis Space & Inductive Bias

Inductive Learning / Prediction :-
We are given examples / data which are of the form

$(\hat{x}, y)$ or $(\hat{x}, f(\hat{x}))$.

where for a particular instance $\hat{x}$ comprises of the values of different features of the instance and y as output.

Types of Inductive Learning problems :-

* for a classification problem = $f(\hat{x})$ is discrete.
* regression = $f(\hat{x})$ is continuous
* probability estimation = $f(\hat{x})$ is probability of $\hat{x}$.

Thus, induction is performed to try to identify a function which can explain the data. ~~This is~~ Unless we see all the possible data points or make a restricted assumption about the language in which hypothesis is expressed or some bias, this problem is not well defined.

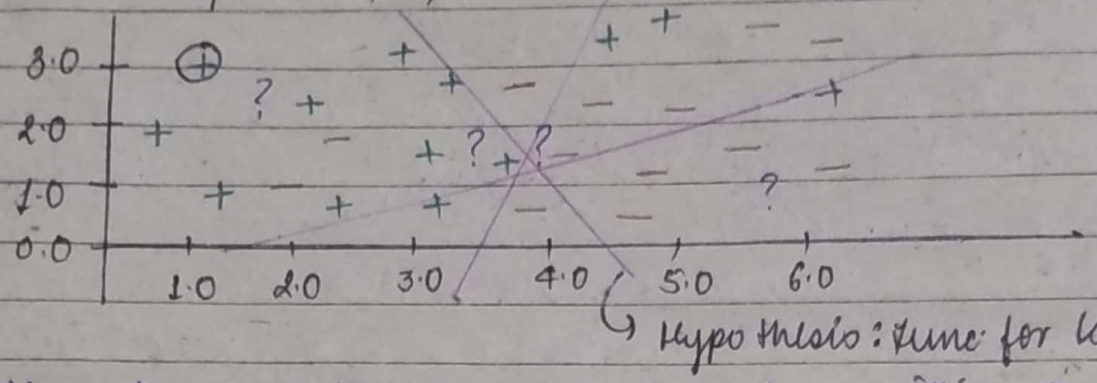
So, it is called an inductive problem.

Instances are described in terms of features. So features are properties that describe each instance in a quantitative manner. For multiple features we defined it as feature vector.

Let us consider two features as x_1 & x_2 that defines 2-dimensional space. N features define n-dimensional space. This is a two class classification

problem where we are given a number of instances or examples, some of which belongs to class 1 and others class 2.

Example: ⟨0.5, 2.8, +⟩ [An instance (x, y), label]



↳ Hypothesis: Func for labelling

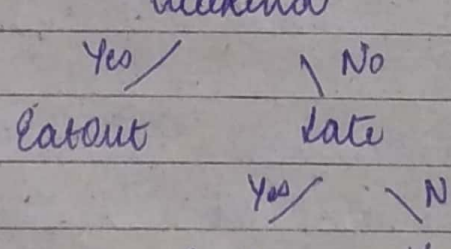
Here the test points are asked to be positive or negative in the prediction problem. For this, we come up with functions (represented with pink line).

Set of All possible legal functions that we come up with during prediction are defined as Hypothesis space (set of legal hypotheses).

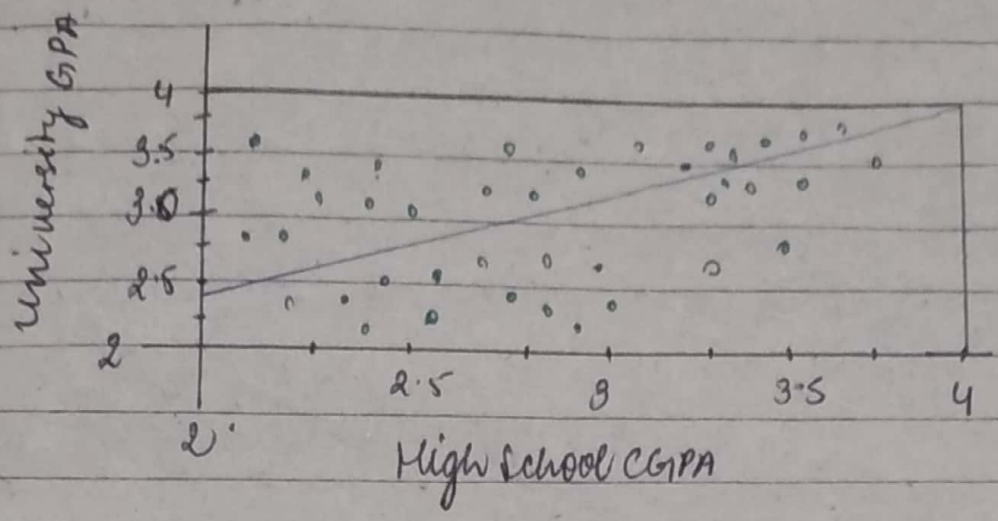
Representation of Functions →

- A function is represented in terms of features and its class (type or language) to define hypothesis space.

1) Decision Tree :-

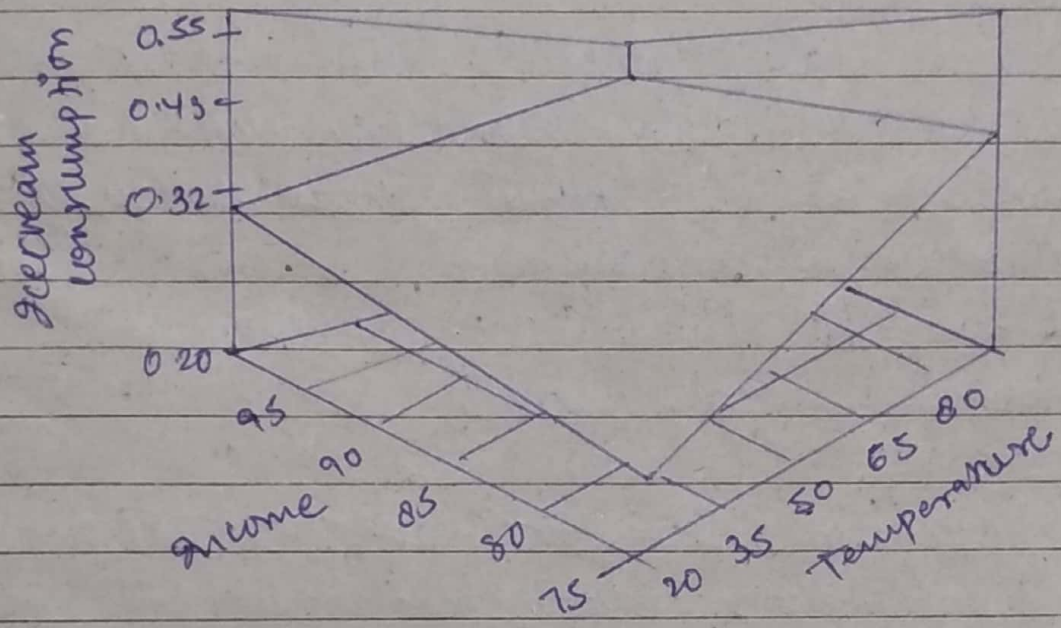


2) Linear functions :-

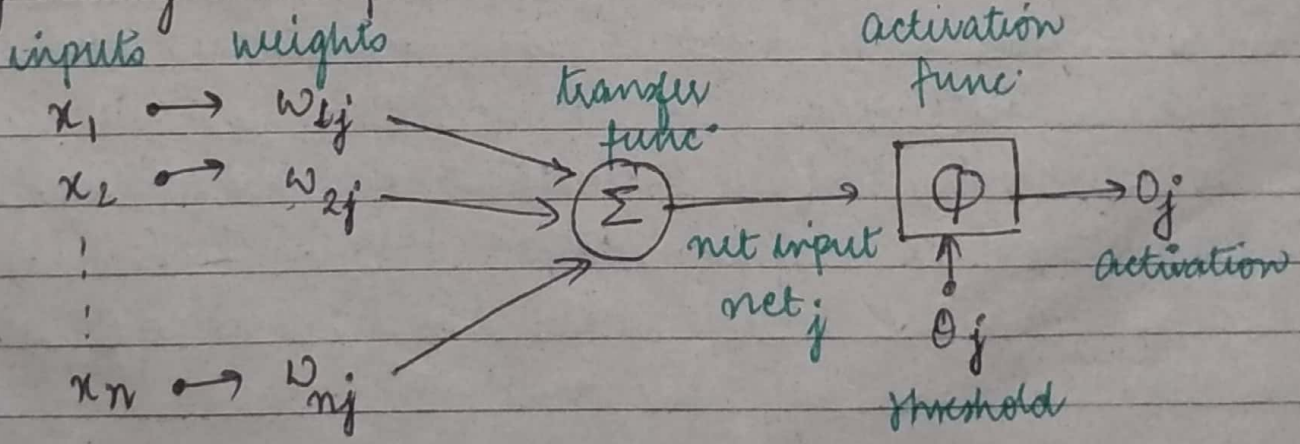


examples.

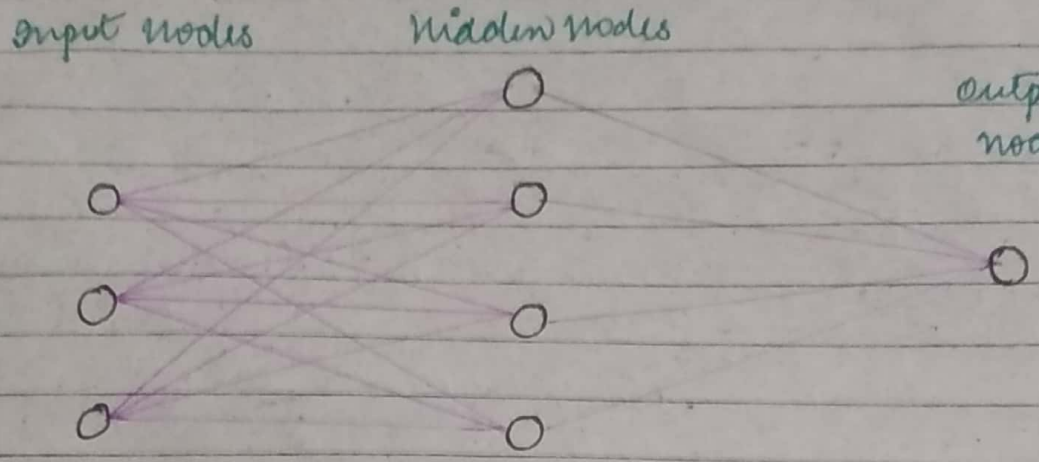
3) Multivariate linear function :-



4) Single Layer Perceptron -



5) Multi-layer neural networks -



Hypothesis space :-

A space of all legal hypothesis that we describe using the features and language that we have chosen. This is a set from which learning algorithm picks a hypothesis.

$R \in H \Rightarrow$ output of learning alg.

$H \rightarrow$ all legal hypothesis

* The space of all hypotheses than can, in principle, be output by a learning algorithm.

* We can think about a supervised learning machine as a device that explores a "hypothesis space".
  - Each setting of the parameters in a machine is a different hypothesis about the function that maps input vectors to output vectors.

classifier / for classification Problem.

* Hypothesis h : Function that approximates f.

* Hypothesis space H :- Set of functions we allow for approximating f.

* The set of hypotheses that can be produced, can be restricted further by specifying a language bias.

* Input :- Training set $S \subseteq X$ (instance space)

* Output :- A hypothesis $h \in H$.

Hypothesis is restricted by defining bias that are of two types :- Constraints
Preferences

* If there are 4(N) input features, there are 2^{16} (2^{2^N}) possible Boolean functions.

* Possible instance is 2^N and Boolean function possible is 2^{2^N} (subsets of func.)

2^{2^N} is also the size of hypothesis space.

* We can put some restrictions on hypothesis space. For this we select a hypothesis language (which may or may not be restricted).
The restricted hypothesis language reflects the

inductive bias of a learner.

Inductive Bias :-

For choosing hypothesis space, we need to make an assumptions and these are two types of assumptions / bias that we can make -

1) Restriction :- (specifing the form of func.)
Limit the hypothesis space example - considering only conjuctive boolean formula except all.
2) Preference :- Impose ordering on hypothesis space.

Inductive learning :-

Inducing a general function from training examples.

- construct hypothesis h to agree with c on the training examples.
- A hypothesis is consistent if it agrees with all training examples. called "consistent hypothesis"
- A hypothesis said to generalize well if it correctly predicts the value of y for novel example (unseen as well as seen data)

Inductive Learning is an Ill Posed Problem :-

Unless we see all possible examples the data is not sufficient for an inductive learning algorithm

to find a unique solution.

Inductive Learning Hypothesis :-

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

Learning as Refining the Hypothesis Space :-

* Concept learning is a task of searching an hypothesis space of possible representations looking for the representations that best fits the data, given the bias.

* The tendency to prefer one hypothesis over another is called a bias.

* Given a representation, data, and a bias, the problem of learning can be reduced to one of search.

Examples of Inductive Bias :-

* Occam's Razor : The simplest consistent hypothesis about the target function is actually the best. (Thus, we prefer the simplest hypothesis here)

* Minimum description length :- when forming a hypothesis, attempt to minimize the length of the description of the hypothesis.

* Maximum margin :- when drawing a boundary between two classes, attempt to maximize the width of the boundary (SVM).

generalization :-

coming up with a function in machine learning is all about generalization.

* Components of generalization error :-

— Bias :- (Restriction or preference in choosing hypothesis)

How much the average model over all training sets differ from a true model ?

* Error due to inaccurate assumptions / simplifications made by the model.

— Variance :-

How much models estimated from different ~~learning~~ training sets differ from each other.

Evaluation and Cross Validation

Experimental Evaluation of Learning Algorithms

* Evaluating the performance of learning systems is important because:-
  Learning systems are usually designed to predict the class of "future" unlabeled data points.

* Typical choices for Performance Evaluation :-
  - Error
  - Accuracy
  - Precision / Recall

* Typical choices for Sampling Methods :-
  - Train / Test sets (Disjoint sets)
  - K-fold Cross-validation.

Evaluating Predictions :-

Suppose we want to make a prediction of a value for a target feature on example x :
- y is the observed value of target feature on example x.

- $\hat{y}$ is the predicted value of target feature on example x.

$\Rightarrow$ $\hat{y} = h(x)$ If $y = \hat{y}$ there is no error.

There are different ways in which error is measured -

For Regression Problems -

1) Absolute Error :- $\frac{1}{n}\sum |h(x)-y|$

2) Sum of squares error :- $\frac{1}{n}\sum_{i=1}^{n} (h(x)-y)^2$

For classification problems -

3) Number of misclassifications :- $\frac{1}{n}\sum_{i=1}^{n} \delta(h(x), y)$

δ is a function which returns 1, when $h(x)$ & y are different and 0 if same.

n = no. of examples of ~~class~~ testing.

Sometimes especially in classification problems it is helpful to define confusion matrix. Suppose, we have a two-class ~~at~~ classification problem and a set of examples on which we are testing.

Hypothesized class \ True class	+	-
+	TP	FP
-	FN	TN
	P	N

~~Some~~ Sum of first column = P

second " = N

The confusion matrix →

TP - The training examples for which true class is positive and we also hypothesized positive.

TN - The training examples where true negative classes are also output as negative by a learning algo.

The learning algorithm also makes mistakes and they are of two types -

FP - The examples are actually negative & our learning algo is wrongly classifying it positive.

FN - The learning algorithm erroriously marks as negative for those examples which should have been positive.

We can have confusion matrix for more than two classes also. The diagonal elements only give correct results in the matrix.

Considering confusion matrix -

The true positive rate is given as -

$$\text{Accuracy} = \frac{TP + TN}{P + N} \qquad \text{Recall} = \frac{TP}{P} \text{ (sensitivity)}$$

$\because P = TP + FN$

$N = FP + TN$

$$\text{Precision} = \frac{TP}{TP + FP}$$

the false positive rate is given as -

$$\text{specificity} = \frac{TN}{TN + FP} = \frac{TN}{N}$$

- We evaluate the learning algorithm on the sample data and the error that we get is called sample error whereas actual error is called the true error.

Sample Error and True Error

* the sample error of hypothesis f with respect to target function c and data sample S is :-

$$error_S(f) = \frac{1}{n} \sum_{x \in S} \delta(f(x), c(x))$$

* The true error (denoted $error_D(f)$) of hypothesis f with respect to target function c and distribution D, is the probability that h will misclassify an instance drawn at random according to D.

$$error_D(f) = P_{x \in D}[f(x) \neq c(x)]$$

* When we hypothesise a particular function, the error

that we get comes from different sources. Due to biased representation, the error that come from limitation in function / hypothesis space.

That we get comes from different sources. The error that come from limitation in representation function / hypothesis space is due to bias inrepresentation.

The error may come because giving the hypothesis space the search algorithm is not exhaustively searching the hypothesis space but only making certain simplification. This is called search bias.

The error may due to the limited size of the sample used for testing, then it is called variance error.

The error may also occurs because of the features / vocalulary we are using is not sufficient to capture everything about the task, then it is called noise.

Difficulties in evaluating hypotheses with limited data

* Bias in the estimate :-
The sample error is the poor estimator of true error
— ⇒ test the hypothesis on an independent test set

* We divide the examples into :-

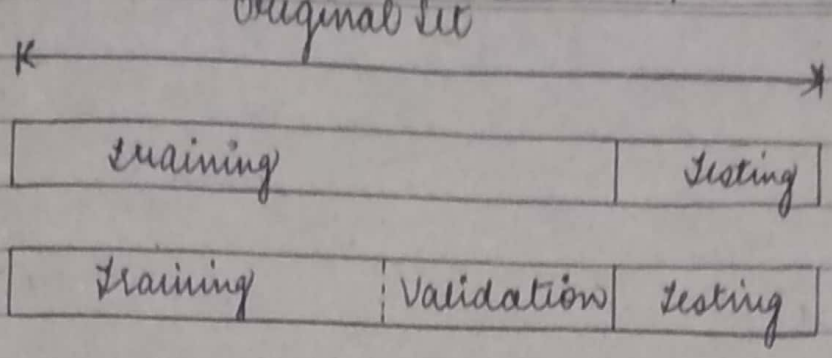
- Training Examples :- that are used to train the learner.
- Test Examples :- that are used to evaluate the learner.

* Variance in the estimate :-
The smaller the test set, the greater the expected variance. The accuracy may be very high or very low on the set.

Way of doing evaluation with limited training data

Suppose we are given some labelled training data. If we use all of them for training we may not really get good estimate of the error as we require an independent set. To remove this problem ideally,
given the labelled training data, we divide it into training set and test set. But division lets the size of training set decrease and as the size of the set becomes too small it gives rise to overfitting type of error. So, we need to use as much of the training examples for training and also we don't want to test on a small test set (variance will be high), for this we evolve a scheme of cross validation.

The standard validation procedure



The data is splitted into training set and testing set. During training when we are tuning the model parameters, we can use some part of training set for validation; after this the output of tuned training and validation is checked in the testing part.

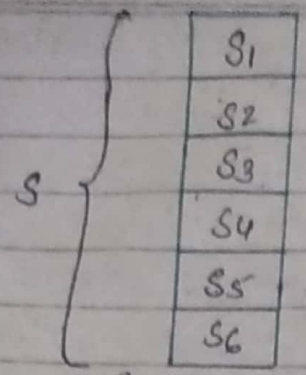
Validation set is used during training to tune the parameters; after the entire training is over, we check the accuracy of the hypothesis on the test set. While splitting data into above three parts — the validation fails to use all the available data.

K-fold cross validation

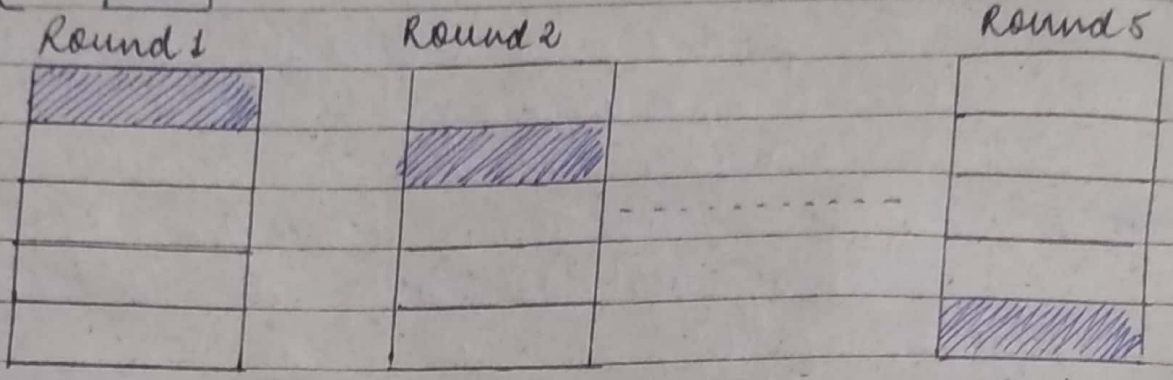
1) Split the data into k equal subsets.

2) Perform k rounds of learning; on each round
   - 1/k of the data is held out as the test set and
   - the remaining examples are used as training data.

3) Compute the average test set score of the k-rounds.



Round i Use S_i for testing
$S - S_i$ for training



validation accuracy 93% 90% 95%

Final accuracy = Avg (Round 1, Round 2 ---)

Trade Off:

* In machine learning, there is always a trade-off between
  - complex hypotheses that fit the training data well as they are flexible but have tendency to overfit the training data and may work poorly on test data special
  - simpler hypotheses that may generalize better; but they may not be able to represent complex functions.
* As the amount of training data increases, the generalization error decreases and we can go for complex hypothesis.

Introduction to ML

* In machine learning programs can be modified for better understanding and applied to various tasks.
* ML analyses, responds and provides a feel if it is a system or human.

* Patterns
  ↓
  Clusters (similar pattern collections)

* Key ~~terminologies~~ — Points :-

1) Data is changed into information. Data can be of any form as image, audio, text, .csv.
2) This is a problem solving tool.
3) It is a combination of computer Science, Engineering and Statistics.
4) Interprets data and acts on it.
5) Optimise performance criteria using past experience.

* Types of ML -

(I) Supervised learning :- teacher oriented learning.
(1) The system previously knows what the output is
(2) going to be.

(II) Unsupervised learning :- We don't have an idea of what the ouput would be. The system first recognise

to the pattern to take out the output.

(III) Reinforcement Learning :- Reward based, feedback oriented learning (from the environment). The programmer decides rewards for certain set of actions. Through this optimality is gained by the system.

* Steps in ML :-

(I) Collect data → comma separated value.
(II) Prepare the input data. • csv file format / text
(III) Analysing the input data such as checking
- patterns
- outliers
- novelity
(IV) Train the algorithm / develop.
(V) Test algorithm
(VI) Use as real world application

* Key Terminologies :-

(I) Expert Systems :- They should be domain specific for a system. They must be understandable, high performance giving, Reliable, High Response; advice given. Example - autonomous car.
Components of expert system →
a) Knowledge Base - keeps knowledge of domains. They are of two types -
(I) Factual Knowledge (collected from journals, notebooks etc)

(II) Heuristic Knowledge –
based on experiments, can be said as art of good guess.

b) Knowledge Representation –
Organising the data.

c) Knowledge acquisation → maintains the quality of knowledge such as completness, information accuracy.

d) Knowledge engineer – the domain experts collections data and codes the data to give to the knowledg -e engineer as information.

e) Inference Engine :-
deducing → deduces the knowledge that the given solution is correct or not.

f) User Interface → interaction between expert system and end user.

(II) Training set :-
Includes collected data.

(III) Validation set :- The portion in one use.
(IV) Test set :- Uses remained data of training set for generalisation using classifier (full-set)

(V) Target variable :- The form in which output is to be saved.