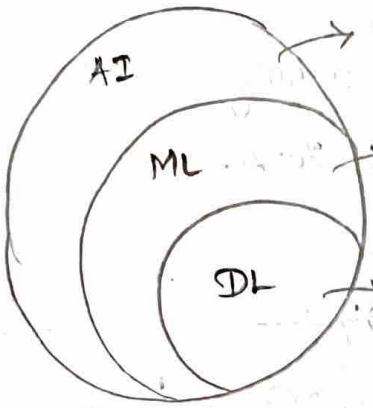
Unit - 1 : Introduction to Machine Learning.

Machine Learning:

* "Field of study that gives computers the ability to learn without being explicitly programmed."
- Arthur Samuel.

Learning:

* "Learning is any process by which a system improves performance from experience."
- Herbert Simon.



- any technique which enables computers to copy human behavior.
- is a AI technique that give computers the ability to learn without being explicitly programmed
- a subset of ML which make the computation of multi-layer neural network feasible

What ML can do?

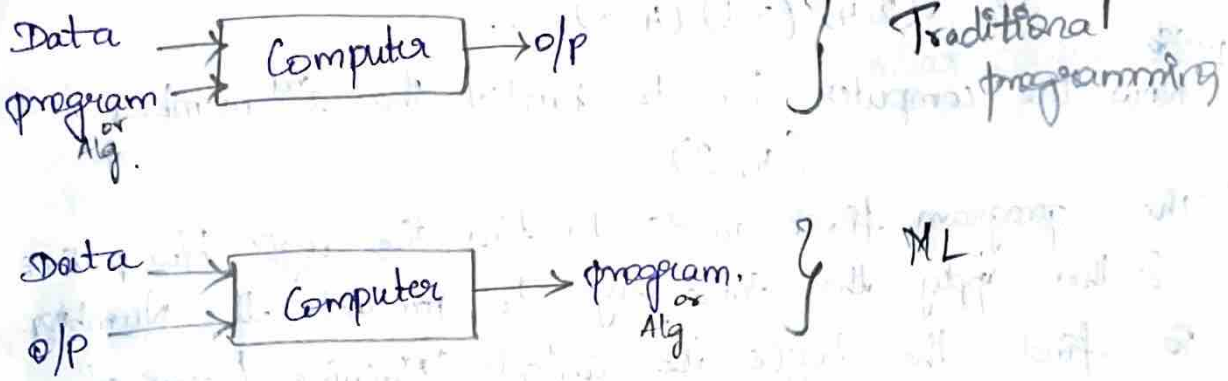
* find which category an object belongs to
  - by "Classification Algorithm".
* find what is strange
  - by "Anomaly Detection Algorithm".
* find how much and how many
  - by "Regression Algorithm".
* find how data is arranged.
  - by "Clustering Algorithm"
* What should I do next
  - by "Reinforcement Algorithm".

ML trends:

- Self driving google car
- Sentiment analysis
- pattern recognition
- fraud detection

- online recommendation
- Web search results

Traditional programming Vs ML :-



How do developer creates a solution ?

1. Developing an alg.
2. Implementing an alg. in code.
3. Usage. [i/p parameters] → [Implemented alg] → [Result]

How data engineer develops a solution using ML ?

1. Data collection & preparation. DB / [Spreadsheet] → [Data]
2. Experimenting with diff alg. to build a better model. [Data] → [ML Alg] → [ML model]
3. Usage [i/p parameters] → [ML model] → [Result]

How do you decide which Alg. to use?

- there are dozens of alg. used. So choosing the right alg. is trial and error method.
- even highly experienced data scientist can't tell whether an alg will work without trying it out.
- But alg selection, depends on size & type of data, insights you want to get from data & how those insights will be used.

Example:

* Imagine you have some sets of pair of numbers.
* Put only 1 number of pair into machine to predict the other half of pair.

i.e, (2, 4) (3, 6) (4, 8) ....

Question:

* Now the computer has to predict the 2nd number for (5, ?)

" The program first needs to find the logic b/w pairs & then apply the same logic to predict the Number. To find the logic it's called "Machine Learning".

Hence after finding the logic it can apply the same logic to predict each number.

ML workflow:

1. Get Data
2. Clean, prepare & Manipulate Data
3. Train Model
4. Test Data
5. Improve.

ML Applications:

✓ Recognizing patterns:
- facial identities or facial expressions.
- handwritten or spoken words
- medical images

✓ Generating patterns:
- generating images or motion sequences (demo)

✓ Recognizing anomalies:
- unusual sequence of credit card transactions.
- unusual sound in car engine.

✓ Prediction:
- future stock prices.

✓ spam filtering, fraud detection.

✓ Recommendation Systems

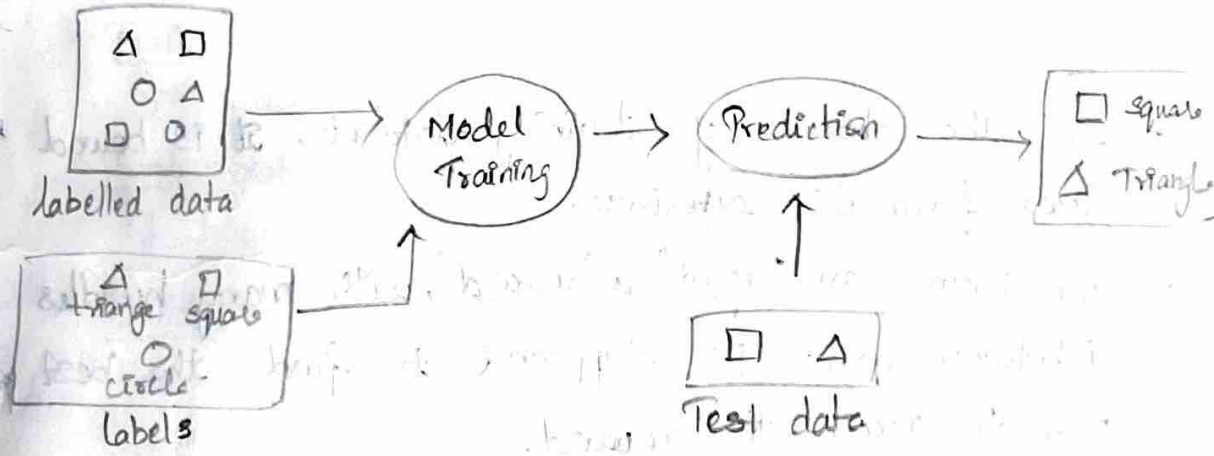
✓ Information Retrieval:
- find doc. or images with similar content.

✓ Data Visualization:
- display a huge DB in revealing way.

Types of ML:

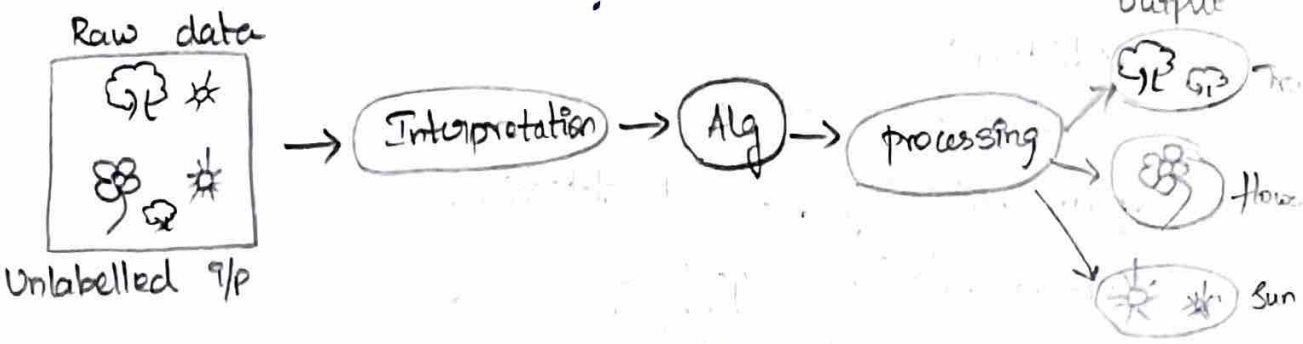
→ Supervised Learning ["Task driven" i.e, predict next value]
→ Unsupervised " ["Data driven" i.e, identify clusters]
→ Semisupervised "
→ Reinforcement " [Learn from mistakes]

① Supervised Learning:
- Its a basic type of ML where ML alg is trained on labelled data.



Eg: Decision Tree, SVM, Naive Bayes, KNN ... etc.

② Unsupervised Learning:
- here labels are not known during training.
- since the labels are not given, hidden structure of data can also be founded.



Eg: PCA, Clustering, association learning.... etc.

③ Semi supervised ~~Unsupervised~~ Learning:

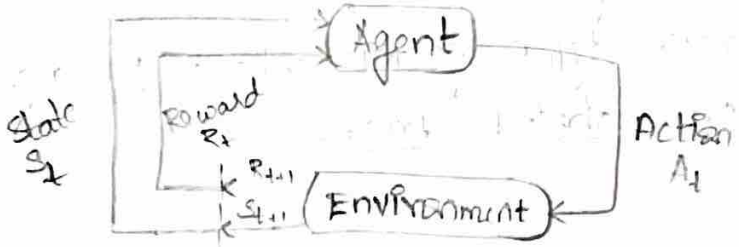
- It combines both labeled & unlabeled examples to generate an appropriate function or classifier.

i.e, its a combination of supervised & unsupervised learning.

- goal is to classify some of unlabeled data using the labeled information set.

Eg: Speech Analysis, Text document classifier etc...

④ Reinforcement Learning:



- In the absence of training dataset, it is bound to learn from its experience.
- we have an "agent" & "reward", with many hurdles inbetween. agent is supposed to find the best possible path to reach the reward.
- favourable outputs are "encouraged" or "reinforced", and non-favourable o/p are "discouraged" or "punished".
- Interpreter will decide whether o/p is favourable or not, if its favourable, then it provides reward orelse it reiterate unit it finds the better result.

Eg: Game playing, Robotics, Business .... etc.

Types: Positive & Negative reinforcement learning.

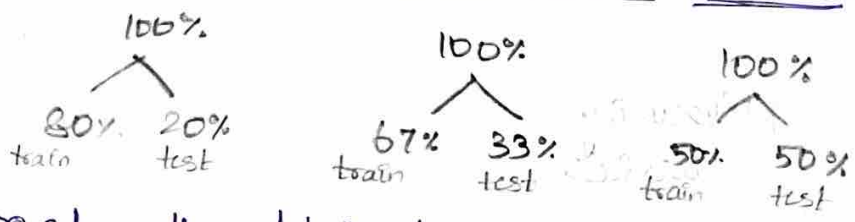
Algorithms:

→ Supervised learning
- Prediction
- classification (discrete labels)
- Regression (real values)

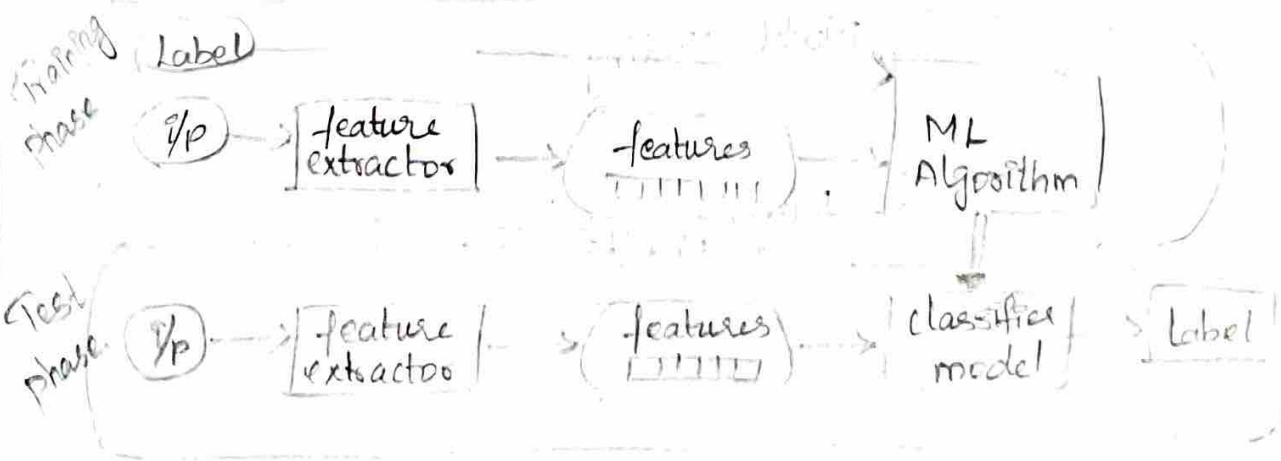
→ Unsupervised learning
- clustering
- probability distribution estimation.
- Finding estimation (in features)
- Dimension reduction.

→ Reinforcement learning.
- Decision making (robot, chess machine)

Training set, Test set and validation set:



Divide the total dataset into 3 subsets:
- Training data is used for learning the parameters of the model.
- Test data is used to get a final, unbiased estimate of how well the networks works.
- Validation set is used to reduce overfitting, tune the parameters - repeat this until we get model with best fit & accuracy.



Hypothesis space:

→ each setting of parameters in machine is a different hypothesis about function that maps i/p vectors to o/p vectors.

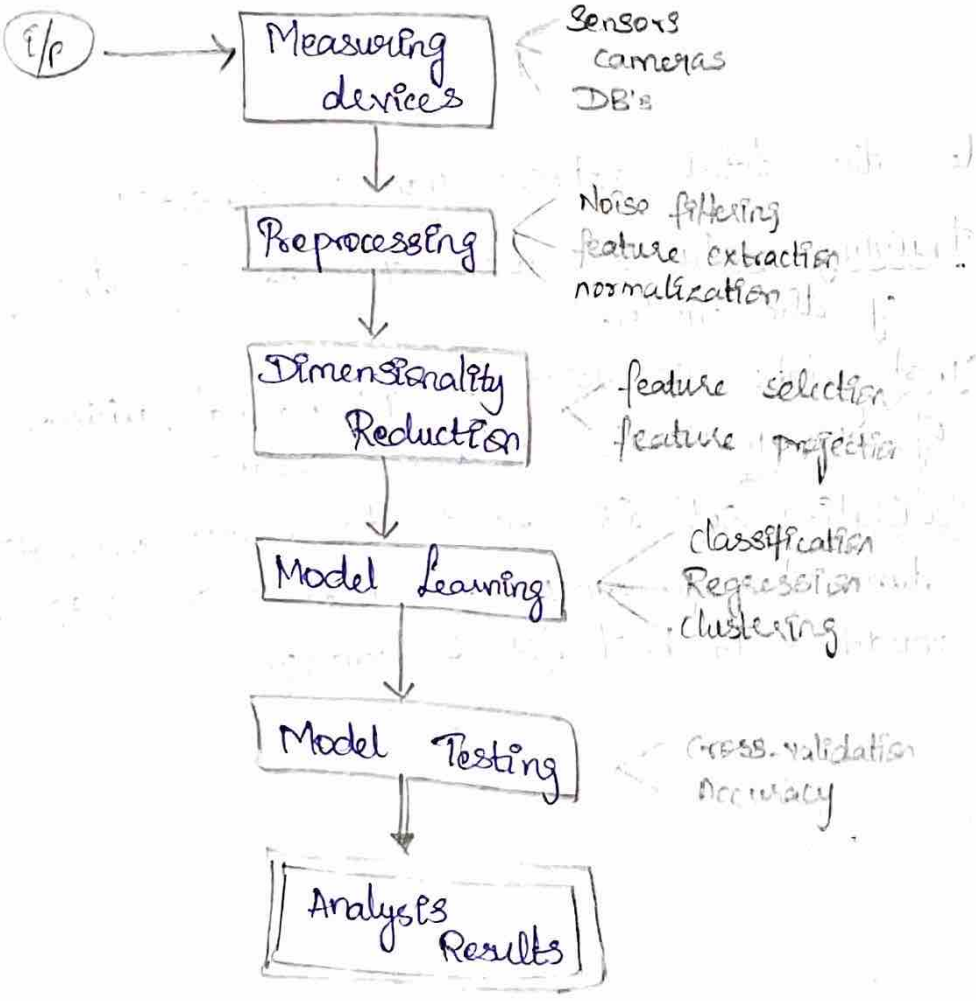
→ If data is noise free, each training example rules out a region of hypothesis space.

→ If data is noisy, each training example scales the posterior probability of each point in the hypothesis space in proportion to how likely the training example is given that hypoth

Thus, the art of supervised ML is in:

- deciding how to represent the i/p & o/p.
- selecting a hypothesis space that is powerful enough to represent the relationship b/w i/ps & o/ps.

Learning Process:

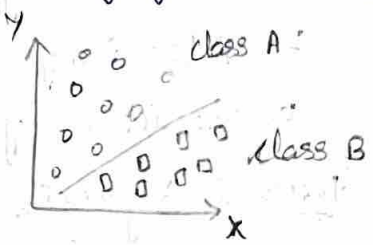


Classification Algorithm:

- is a supervised learning technique that is used to identify the category of new observations on the basis of training data.
- It learns from given dataset & then classifies new observation into number of classes or groups, such as yes or no, 0 or 1, Spam or not spam etc....
- classes can be said as target or labels or categories
- unlike regression, o/p variable is category not a value.

$y = f(x)$

Categorical o/p i/p variable



Types of classifications:

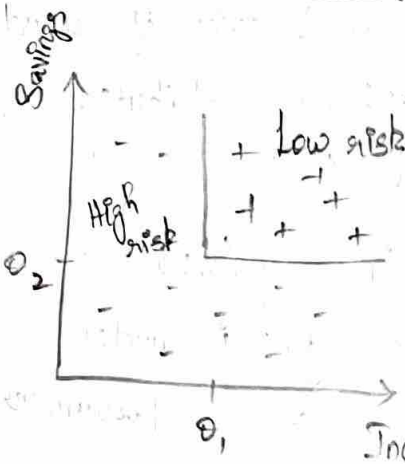
* Binary classifier
  - has only two possible outcomes.
  - Eg: male-female, Yes-No, 0-1, ... etc.
* Multi-class classifier
  - has more than two possible outcomes
  - Eg: classification of types of crops
    " " " music ... etc.

Types of classification Alg:

→ Linear Models
  - ↳ Logistic Regression
  - ↳ SVM

→ Non-linear Models
  - ↳ KNN
  - ↳ Naive Bayes
  - ↳ Decision Tree
  - ↳ Random forest

Credit Scoring



If income $> \theta_1$ & saving $> \theta_2$
then low risk
else
high risk

Regression Analysis:

- Is used when o/p variable is continuous values such as 'weights', age, dollars etc...
- Is a supervised learning technique which helps in finding correlation b/w variables & enables to predict the continuous o/p variable based on one or more predictor variable.
- mainly used for prediction, forecasting, time series etc.

Examples:

↳ prediction of rain using temperature & other factors
↳ Determining market trends.
↳ prediction of road accidents due to rash driving

Types of Regression:

- Linear regression
- Logistic "
- Polynomial "
- Support Vector "
- Decision Tree "
- Random forest "

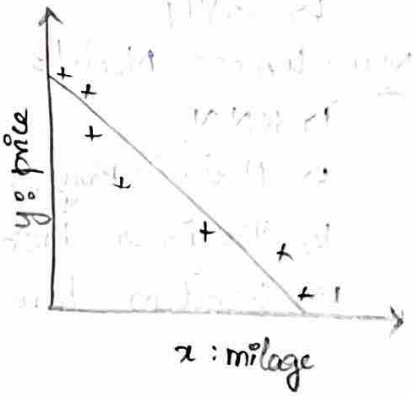
Prediction - Regression:

Predict price of used car

x: car attributes
y: price

$$y = g(x|\theta)$$

where $g()$ is model
θ is parameter



Learning Associations:

- Basket Analysis:
  $P(Y|X)$ is a probability that somebody who buys X also buys Y,
  where X, Y are products, services...

Eg: $P(\text{chips}|\text{beer}) = 0.7$
i.e 70% of customers who buys beer also buy chips.

- Find frequent elements/items.
- " the associations b/w them (or) relation b/w them.

Eg: Single Dimensional Association rule:
Buys ('x', 'computer') → Buys ('x', 'c/w')
Multi-Dimensional Association rule:
Age ('X', '40-60') and Income ('X', '50-to lakhs') → Buys ('X', Computer)

i.e Imagine that you are a sales manager at xyz company & you are talking to a customer who recently bought a PC & camera from store. What should you recommend to him/her next?

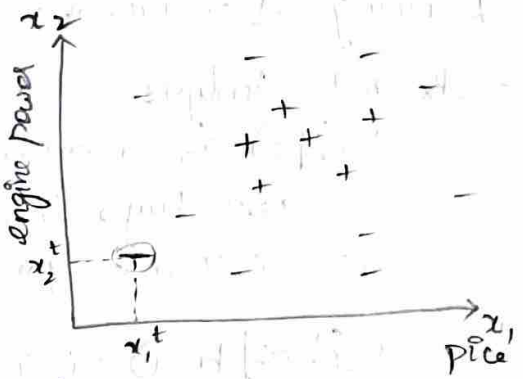
↳ Frequent patterns & association Rules are the knowledge that you want to mine in such scenario.

Algorithms:
→ Apriori
→ Frequent pattern growth tree.

Supervised Learning - Learning a class from Examples:

* Lets take a set of cars. i.e, class - C: "family of cars"
  Cars
  - A group of people look at the cars & label them as family car or not with two attributes as price & engine power.
  - ignore other attributes such as color, seating capacity etc
  - The cars that they believe as family cars are "positive examples" & other cars are "negative example".

- Single data point corresponds to one sample car.
- Coordinates :
  x_1 = price
  x_2 = engine power



- '+' → positive example (family car)
  '−' → negative " (not a family car)

Training Set :

→ Variables 'x' and 'r'.

where $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ → price, → engine power

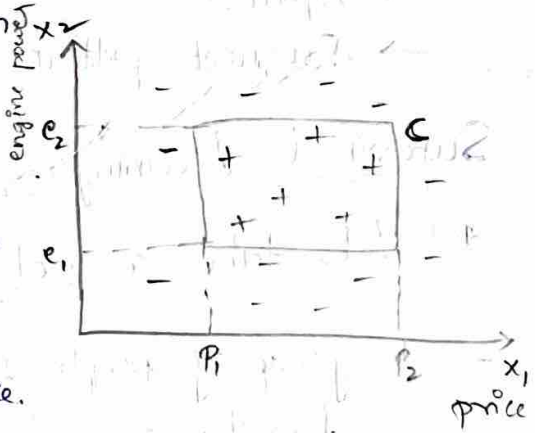
result set $r = \begin{cases} 1, & \text{if } x \text{ is +ve example (family car)} \\ 0, & \text{if } x \text{ is −ve example (not family car)} \end{cases}$

→ Each car is represented by ordered pair (x, r) and training set contains 'N' such examples.

i.e, $\boxed{X = \{x^t, r^t\}_{t=1}^{N}}$ where t is training set

Hypothesis class :

- If a car is family car, then its price & engine power should be in certain range.



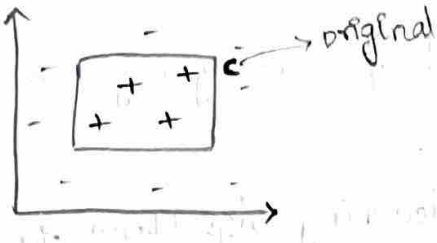
i.e, $(p_1 \leq \text{price} \leq p_2)$ and $(e_1 \leq \text{engine} \leq e_2)$

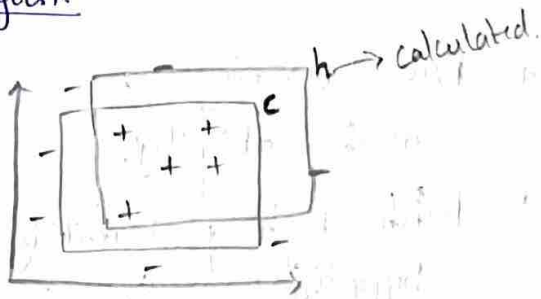
- the class of family car is a rectange in price-engine power space.

- Hypothes h is specified by quadruple of (p_1, p_2, e_1, e_2) to approximate 'C'.

$$h(x) = \begin{cases} 1, & \text{if } h \text{ classifies } x \text{ as +ve example (family car)} \\ 0, & \text{if } h \text{ " " −ve " (not a family} \end{cases}$$

- 'C' is a target function.
- In real life we don't know C(x), so we cannot evaluate how well h(x) matches C(x).
- Instances within rectangle represents family car and outside are not family cars.
- there may be some error region.





False positive & false Negative:

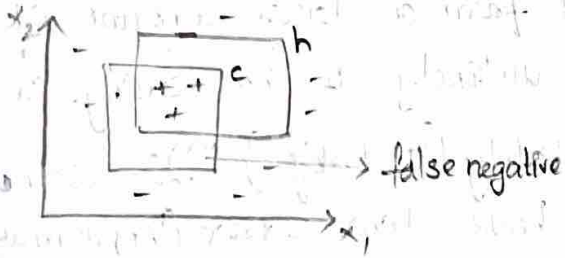
→ 'C' is actual class & 'h' is our induced hypothesis.

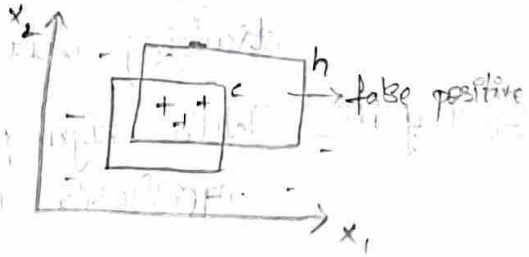
→ point where C is 1 & h is 0 is false negative

→ " " C is 0 but h is 1 is false positive.

→ true positive & true negative - are correctly classified.

	P	N
P	TP	FP
N	FN	TN



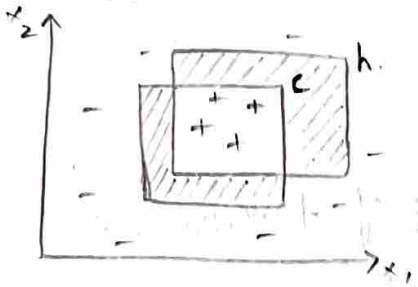


Error:

- the empirical error is the proportion of training instances where predictions 'h' do not match required values given in X.

$$E(h|X) = \sum_{t=1}^{N} 1(h(x^t) \neq r^t)$$

$h(x) = \begin{cases} 1, & +ve \\ 0, & -ve \end{cases}$ $r = \begin{cases} 1, & +ve \\ 0, & -ve \end{cases}$



- probability of error should be small.

Error region : $P(C \text{ xor } h) \leq \varepsilon$

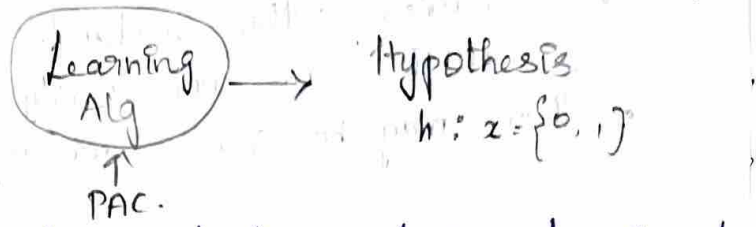
Supervised Learning : PAC Learning :-

↳ Probably Approximately correct (PAC).

- Cannot expect the learner to learn a concept exactly

x	150	170	165
class	0	1	0

training set



* PAC learning is a framework for analysing learning alg. mathematically.
* With high probability, a PAC learning algorithm find a hypothesis that is approximately identical to the hidden target concept.

✓ Given training set of labelled example, learning alg generates a hypothesis. Run hypothesis on test set to check how good it is.

✓ Intuition : A hypothesis built from a large amount of training data is unlikely to be wrong. i.e PAC

✓ Goal of PAC : With high probability ('probably') the selected hypothesis will have low error ('Approximately Correct')

* PAC learning requires,
  - Small parameters ε and δ.
  - with probability atleast $(1-\delta)$, a system learn the concept with error at most ε.

Notations :

1) Instance space x.
2) Concept class, C - family of functions, 'c'
   $c : X \to \{0,1\}$
3) Hypothesis class, H - family of functions, 'h'
   $h : X \to \{0,1\}$
4) Probability distribution, f - training samples are generated randomly from x according to f.

5) Learning Alg, A. or L

Defn:

A concept class 'C' is PAC learnable if, there is an alg. 'A' which for samples drawn with any probability distribution 'F' and any concept $c \in C$, with high probability produce a hypothesis $h \in C$ where error is small

Example:

same from last topic Hypothesis space to error

Error region: $P(c \text{ XOR } h) \leq \varepsilon$

Approximately Correct:-

- the hypothesis h, that approximately correct and error is less than or equal to ε.

where $0 \leq \varepsilon \leq 1/2$ i.e, $P(c \text{ XOR } h) \leq \varepsilon$

Probably Approximately Correct:

- less error with high probability

$[P(\text{Error}(h) \leq \varepsilon)] \leq 1 - \delta$

$[P(P(c \text{ XOR } h) \leq \varepsilon)] \leq 1 - \delta$

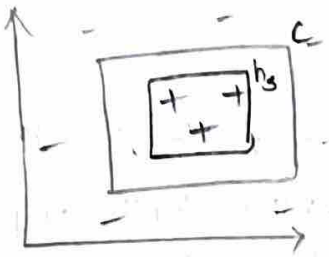
Formal defn:

A concept class, C is said to be PAC learnable by alg., L if for all $c \in C$, distribution F over X, ε such that $0 < \varepsilon < 1/2$ and δ such that $0 < \delta < 1/2$, the learner L will output a hypothesis 'h', with probability atleast $(1-\delta)$ and $\text{error}_F(h) \leq \varepsilon$.

PAC learnability for axis-aligned rectangle:

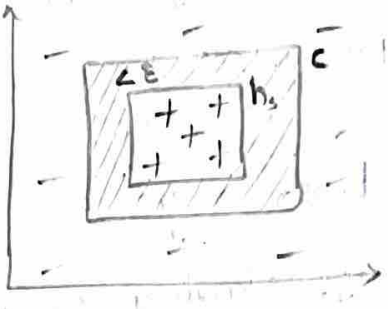
↳ Specialization:

- h_s is the tightest possible rectangle around a set of positive training examples.
- h_s is subset of C, hence error region = c-h

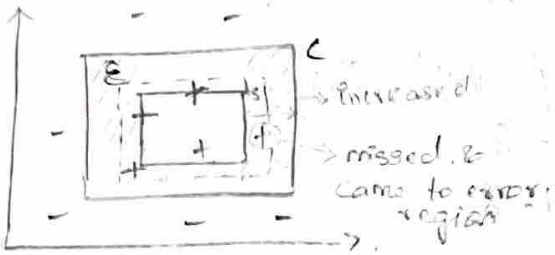


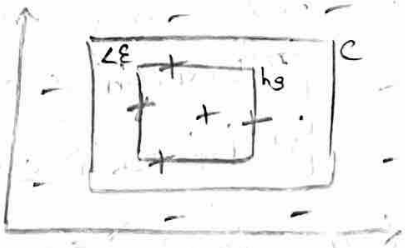
Approximately Correct:

- If an hypothesis lies b/w h & c (shaded region) then it is approximately correct.



- If the generated hypotheses does not touch any of these region, then error region is greater than ε and not approximately correct, Cox error region got increased.
- Should have atleast one +ve example at each side of rectang





Error Region:

Error region = Sum of four rectangular strips $< \varepsilon$

Each strip is atmost $\varepsilon/4$

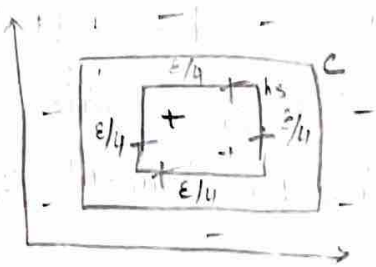
prob. of +ve example falling in any one of strip $= \varepsilon/4$

prob. of randomly drawn +ve example misses a strip $= 1 - \varepsilon/4$

P (m instance miss a strip) $= (1 - \varepsilon/4)^m$

P (m instance miss any strip) $< 4(1 - \varepsilon/4)^m$

$\therefore$ $\boxed{m > 4/\varepsilon \log 4/\delta}$



Ex. problem 1:

Hypothesis h1 generated the errors with respect to price & engine power of given 10 samples,

Given: $\varepsilon = 0.05$, $\delta = 0.20$

S No	Error(h1)
1	0.001
2	0.025
3	0.07
4	0.003
5	0.035
6	0.045
7	0.027
8	0.065
9	0.012
10	0.036

Now, $P(h1) \geq 1 - \delta$

$P(h1) = 8/10$ → total 10 values. where 3rd & 8th are greater than ε so [10 - 2 = 8]

$= 0.80$

$\therefore 0.80 \geq 1 - 0.20$

$0.80 \geq 0.80$

Hence h1 is probably approximately correct.

Ex. problem 2:

Hypothesis h2 generated the errors with respect to price & engine power of given 10 samples,

Given: $\varepsilon = 0.05$, $\delta = 0.20$

S.No	Error(h2)
1	0.012
2	0.015
3	0.071
4	0.063
5	0.022
6	0.045
7	0.011
8	0.029
9	0.066
10	0.034

$P(h2) \geq 1 - \delta$

$P(h2) = 7/10$ 3rd, 4th, 9th are greater than ε

$= 0.70$ [10 - 3 = 7]

$0.70 \geq 1 - 0.20$

$0.70 < 0.80$

Hence h2 is not probably approximately correct

Noise and Model Complexity:

→ 'Noise' is any unwanted anomaly in the data & due to noise, the class may be more difficult to learn and zero error may be infeasible with simple hypothesis class.

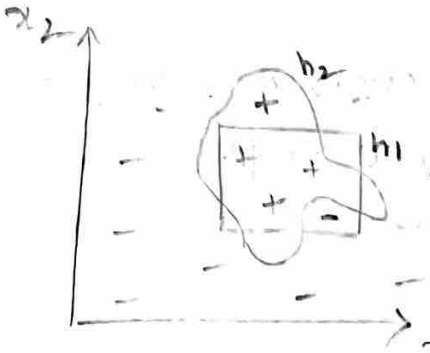
Reason for noise:

- There may be imprecision in recording the i/p attributes, which may shift the data points in i/p space.
- There may be errors in labeling the data points, which may relabel positive instance as negative & vice versa. This is called as 'teacher noise.'

- There may be additional attributes, which we have not taken into account, that affect the label of instance. Such attributes may be hidden or latent which may be unobservable.
  ↳ the effect of these neglected attributes is thus modeled as random component & included in 'noise'.

Effect of noise:

→ Noise distorts data.
→ due to noise, the learning problem will not give any accurate results



Model complexity

h_1 → needs only 4 points to form hypothesis since its a rectangle
h_2 → requires more points cox its freely formed hypothesis

[Here h_2 give more accurate result than h_1]

Also h_1 has one -ve sample & misses the -ve sample but h_2 contains only +ve sample

✓ When there is noise, there is not a simple boundary between the +ve & -ve instances, & zero misclassification error may not be possible with simple hypothesis.
✓ A rectangle (h_1) is a simple hypothesis with 4 parameters defining the corners.
✓ A arbitrary closed form can be drawn by piecewise functions with larger number of control points.

Occam's Razor:

Using simple rectangle makes more sense cox of following:

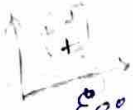
1. It is simple model to use.
   ↳ easy to check whether point is inside or outside rectangle.

2. It is simple model to train & has fewer parameters.
   ↳ easy to find corner values of rectangle than the control points of arbitrary shape.
3. It is simple model to explain.
   ↳ a rectangle simply corresponds to defining intervals on two attributes. By learning a simple model, we can extract info. from raw data given in training set.
4. Given the empirical error, we say that a simple model would generalize better than complex model.
   ↳ this principal is known as 'occam's razor', which states that simpler explanations are more plausible and any unnecessary complexity should be shaved off.

Learning Multiple classes:

→ Single class problem:

- Considering only single class while learning the example are called as single class problem.



Eg: learning a family car, we have +ve example belonging to class family car & never consider other samples

→ Two class problem:

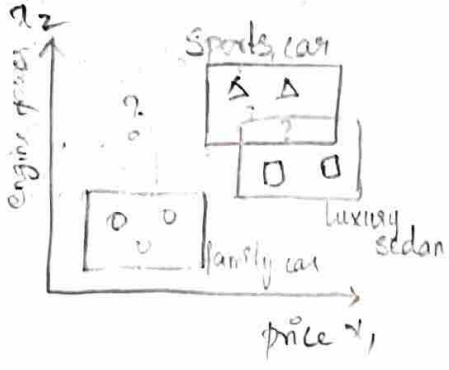
- Considering two classes while learning the example are called two class problem.

Eg: Learning a family car, we have +ve examples belonging to class family car & -ve examples belonging to all other cars.

→ Multiple classes:

- More than 2 classes are said to be multiple classes

Eg: Consider three classes:
- family car
- Sports car
- luxury sedan.



- Here three hypotheses induced, each one covering instance of one class & leaving outside the instance of other two classes. (like when P_1 & P_2 are two some value then it is family etc)
- And '?' are reject regions where 'no' or more than one class is chosen.

* Here K classes denoted as,

'C_i' where $i = 1, 2 \ldots K$.

an o/p instance belong to one & exactly one of them.

$$X = [x^t, r^t]_{t=1}^{N}$$

where r has K dimensions &

$$r_i^t = \begin{cases} 1, & \text{if } x^t \in C_i \text{ (belong to any one class)} \\ 0, & \text{if } x^t \in C_j, j \neq i \text{ (doesn't belong to any class)} \end{cases}$$

i.e, reject region

* The boundary separating the instance of one class from the instances of all other classes. (from hypothesis)
* $K = 2$ if it has two classes like +ve & -ve classes so it is K-class classification problem.
* The training set belonging to C_i are +ve instance of hypothesis 'h_i' & the examples of all other classes are -ve instance of 'h_i'.
* So, in K-class problem., we have K-hypotheses to learn,

$$h_i(x^t) = \begin{cases} 1, & \text{if } x^t \in C_i \\ 0, & \text{if } x^t \in C_j, j \neq i \end{cases}$$

Error:

The total error takes the sum over prediction for all classes over all instances.

$$E\left(\{h_i\}_{i=1}^{K} \mid X\right) = \sum_{t=1}^{N} \sum_{i=1}^{K} 1\left(h_i(x^t) \neq r_i^t\right)$$

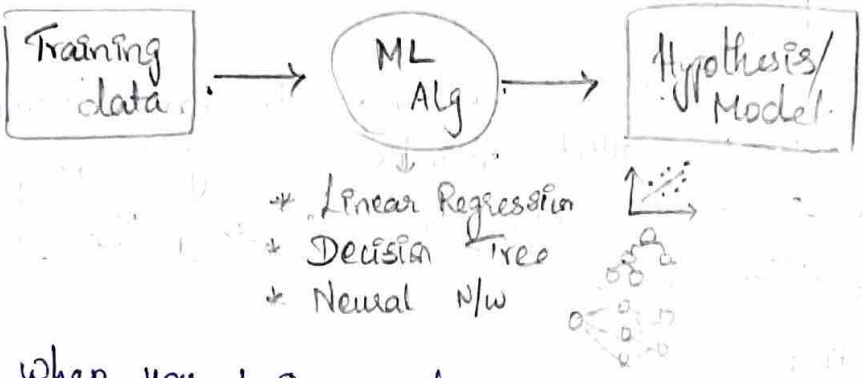
K - classes, t - No. of training samples.

Eg: Handwritten digit recognition dataset,
all digits has similar distribution to belong to same class

0 0 0 0 → C1
1 \ / 1 → C2
2 2 2 2 → C3
⋮ ⋮

Model Selection and Generalization:

Model:



- when you train an alg. with data it will become a model.

Model selection:

→ is a process of selecting an optimal model from a set of candidate models, given data.

Data used in learning are:
- ↳ Training data
- ↳ Validation data
- ↳ Test data.

→ finding an optimal model is minimizing both the bias & variance
(bias: error of train data; variance: error of test data)

Bias:

↳ can be defined as diff. b/w the expected prediction of our model & correct value which we try to predict

↳ High bias can cause our model to miss significant relations b/w our features (X) & o/p's (Y) so it cannot learn training data.

↳ This is also known as 'under-fitting'.
(make lot of assumptions which cause inaccurate predictions)

Variance:

↳ is the error from sensitivity to small fluctuation in the training data.

↳ high variance can cause random noise to be introduced into training data rather than intended o/p's..

↳ This is also known as "over-fitting."

learns training data too well & cannot generalize to new data.

Inductive bias:

- set of assumptions we make to have learning possible is called inductive bias of learning alg.
- model selection is also about choosing the right inductive bias.

Generalization:

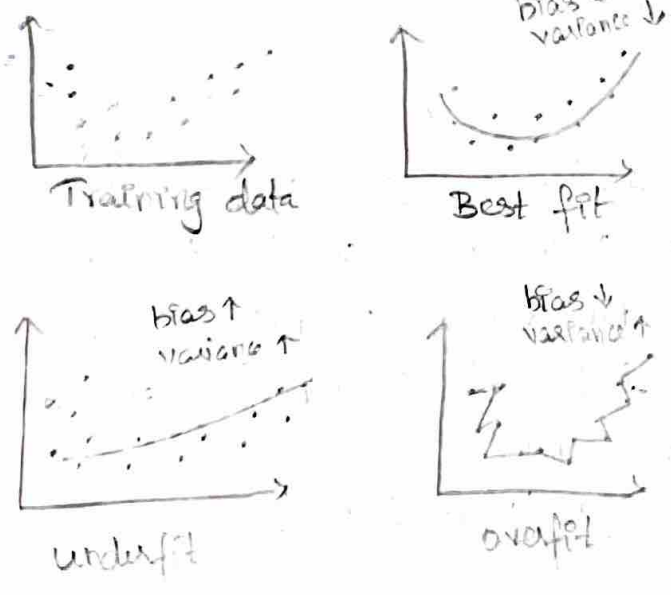
- is how well a model trained on training set predicts the right o/p for new instances.
- the model should be selected having the best generalization.

Need to avoid two problems:

↳ underfitting.

↳ overfitting.

Example: Linear Regression



Classification problem

Underfitting:

- production of ML model that is not complex enough to accurately capture relationships b/w dataset features & target variables.

Overfitting:

- production of model which corresponds exactly to dataset (which may contain noise) & therefore fail to predict new data

---

Dimensions of Supervised ML Algorithms:

* we know that,

$$\mathcal{X} = [x^t, r^t]_{t=1}^{N}$$

where, t is training set with 1 to N samples.
x^t is arbitary i/p
r^t is desired o/p. (is a real value in regression)

o/p for two class learning
for (K > 2) multi class, exactly 1 dimension of r^t is 1 & other is 0

* The aim is to build a good & useful approximation to r^t using the model $g(x^t|\theta)$ parameters

* For doing this, three decisions must be made:
  1. Model we use in learning
  2. Loss function.
  3. optimization procedure.

⇒ Model used in learning:

$$g(x|\theta)$$

where $g()$ is model
x is i/p (sample)
θ are parameters

→ Loss function:

→ If there is a error, there will be a difference b/w desired o/p 'r^t' & can be found using the loss function $L()$.

$$E(\theta|\mathcal{X}) = \sum_t L(r^t, g(x^t|\theta))$$

- approximation error (or) loss is the sum of losses over the individual instances.

where $E(\cdot)$ is error
θ is parameters
x is sample
t is training set from 1 to N.
$L(\cdot)$ is loss function.
r^t is o/p.
$g(x^t|\theta)$ is model.

→ optimization procedure:

- is used to find θ^* that minimizes total error.

$$\theta^* = \arg\min_{\theta} E(\theta|X)$$

where,
arg min → returns argument that minimizes.
$E(\theta|X)$ → error.
parameter, sample.