History →

i) first Machine learning System was developed in the 1950's.

ii) 1952, Arthur Samuel was at IBM developed a program for playing Checkers.

A/c to Samuel → "It is a field of study that gives computers the ability without being explicitly programmed."

1960's

→ Neural N/W → Rosenblatt's perceptron.
→ Pattern Recognition.
→ Minsky & papert proved limitation of perceptron.

1970's

→ Symbolic Concept induction
→ Expert systems and knowledge acquisition bottleneck
– Quinlan's ID3
– Natural language Processing.

1980's

→ Advanced decision tree and rule learning
→ learning and planning and problem solving
→ Resurgence of Neural N/W
→ Focus on Experimental Methodology.

90's ML & Statistics –

→ Support vector Machines
→ Data Mining
→ Adaptive agent & web Applications.
→ Text learning.
→ Reinforcement learning
→ Ensembles
→ Bayes Net learning

1994 → self driving Car road test
1997 → Deep blue beats Gary Caspar

Popularity of this field in recent time and the reason behind that —

→ New Software / Algorithm
- Neural Network
- Deep learning

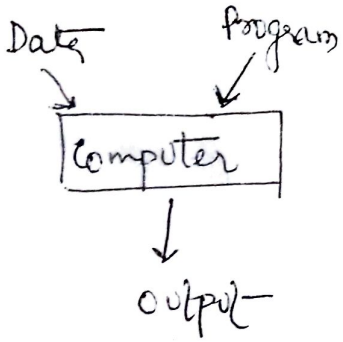
— New Hardware
- GPU's

— Cloud Enabled
— Availability of big data

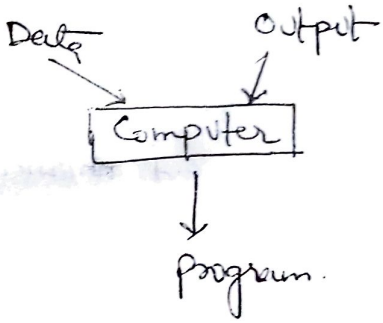
2009 - Google builds self driving Car
2011 - Watson wins Jeopardy
2014 - Human Vision surpassed by ML systems.

How Machine learning solutions differs from Programatic sol^n →

Algorithm



ML



Algorithm → The programs takes data as input and the program produces output.

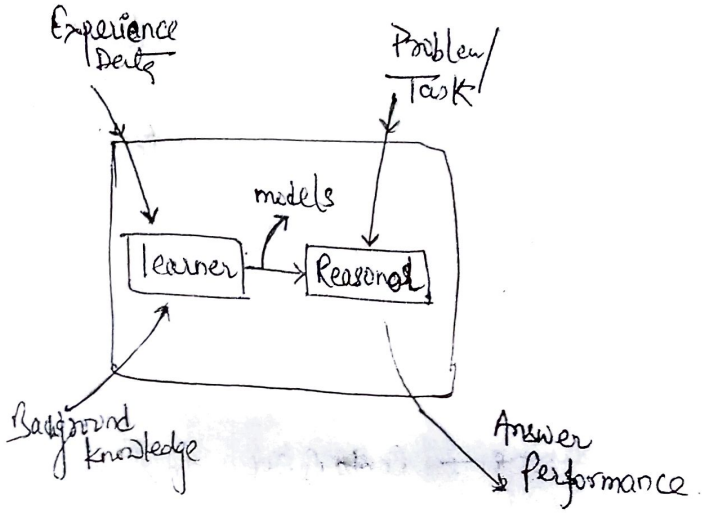
ML → you are feeding the data as input as well as output data and you are getting a program or a model with which you can solve subsequent task

ML → learning is the ability to improve one's behavior with experience so, it is about building computer system, that automatically improves with experience and we have to discuss what are the fundamental laws that governed the learning processes.

Machine learning explores algorithm that learn from data, build Models from data and this model can be used for different tasks.

exm - prediction, decision Making or solving tasks.

Tom Mitchell → defⁿ → A Computer program is said to learn from experiance E with respect to some task T and performance Measure P, if its performance at tasks in T, as measured by P, improves with Experiance E.



Domain & 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 — use historic medical records to learn which future patients will respond best to which treatment

Vision :-

Robot Control
NLP
Speech recognition
Medical Transaction
Financial

→ features

1. Choose the training experiance
2. Choose the target function (that is to be learned) → Hypothesis language
3. Choose how to represent the target function class of fn on features.
4. Choose a learning algorithm to infer the target-fn.

Types of Learning →

- Supervised learning → o/p (labeled training data)
  i/p → (de-classified training examples)
  - Given an observation x, what is the best label for y?

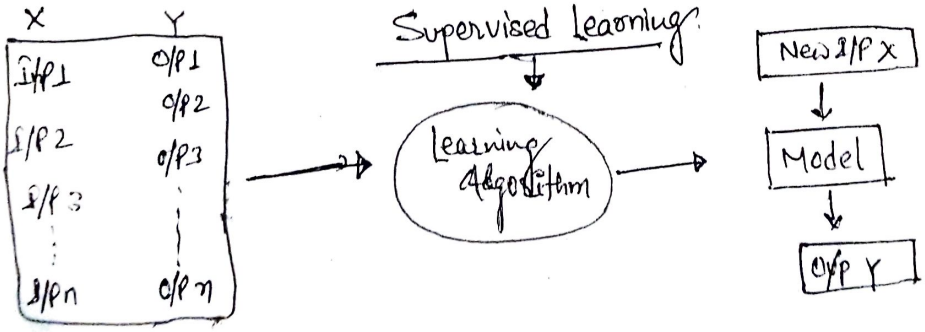
Unsupervised learning — (unlabeled training data).

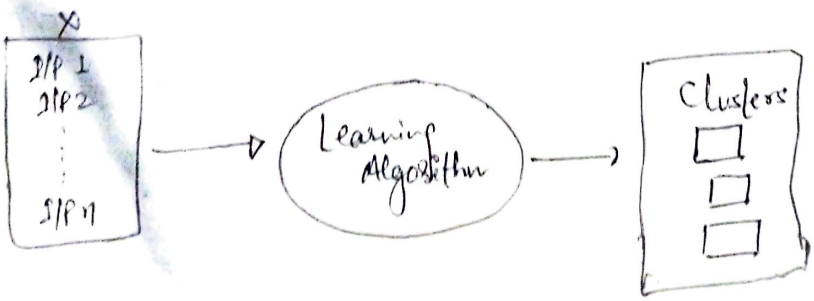
- x
- Given a set of x's, cluster or summarise them

Reinforcement learning —

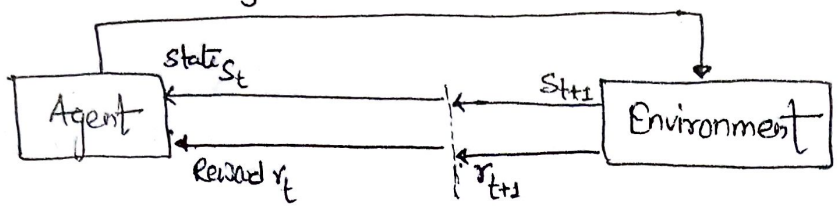
- Determine what to do based on rewards & punishments.

Semi Supervised Learning — Combo of supervised & unsupervised learning.





Reinforcement learning →



Supervised learning →

(input feature) $X_1, X_2 \dots X_n$ | Y (target feature)

- discrete value feature (classification)
- Continuous value feature (Regression)

	$X_1, X_2 \dots X_n$	Y
I_1	$a_1, a_2 \dots a_n$	Y_1
I_2	$b_1, b_2 \dots b_n$	Y_2
I_3	$c_1, c_2 \dots c_n$	Y_3

Test instance
$Z_1 Z_2 \dots Z_n$

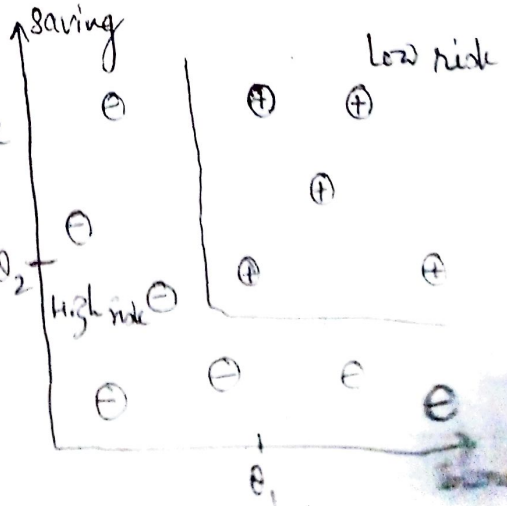
- A set of input features $X_1, X_2 \dots X_n$.
- A target feature Y
- A set of training exm where the values for the input features and the target features are given for each exm.
- predict the values for the target features for the new exm.
  - Classification when Y is discrete
  - Regression when Y is Continuous.

Classification →

Exm → Credit Scoring

Differentiating between low risk & high risk customers from their income & Savings

Discriminant :- if income $> \theta_1$ AND Savings $> \theta_2$
Then low-risk
else High risk



Regression :—

Price of used car

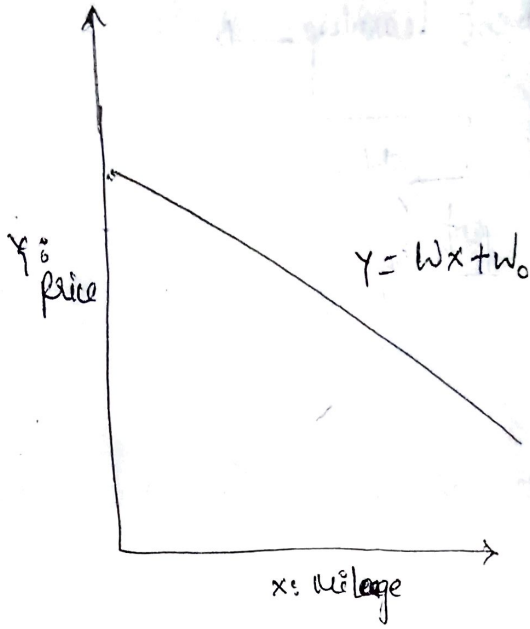
x : car attributes

Y : price

$Y = g(x, \theta)$

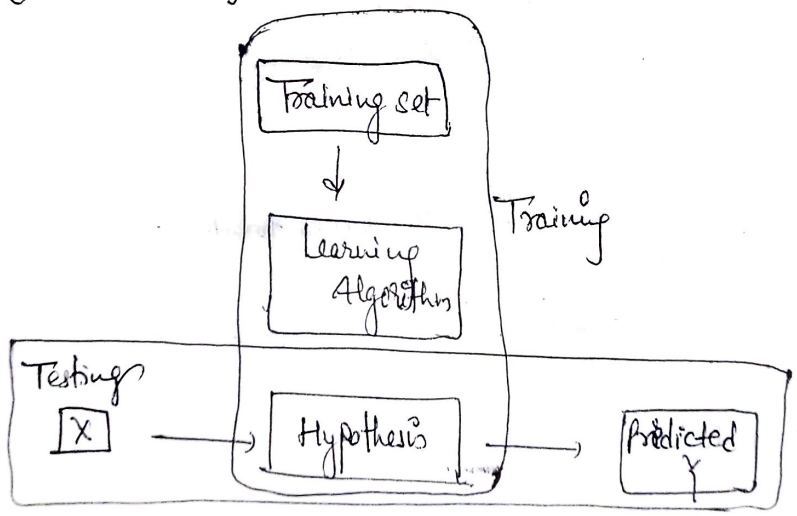
g() Model

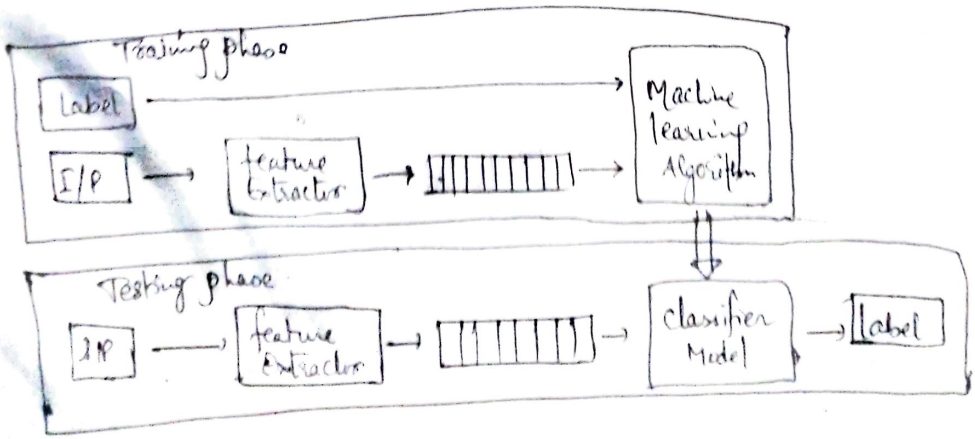
θ parameters



The individual observations are analyzed into a set of quantifiable properties. which are called features.

→ Categorical ("A", "B", "O", blood group).
→ ordinal ("large", "small")
→ Integer-valued (Age)
→ Real valued (height)



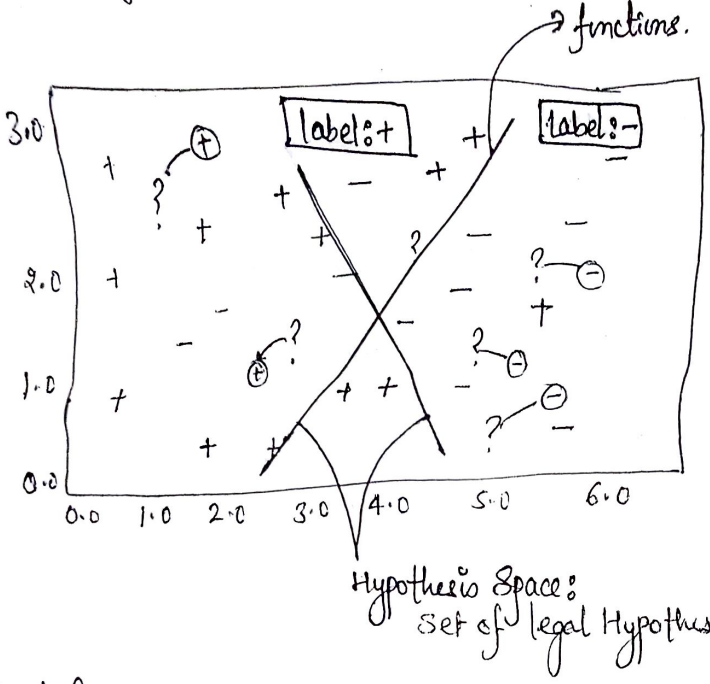


Given eqn $(\hat{x}, y)$, $(\hat{x}, f(x))$

Classification — $f(x)$ → discrete

Regression — $f(x)$ → Continuous

Probability of estimation — $f(\hat{x})$ = probability of $\hat{x}$.



Hypothesis Space:
Set of legal Hypothesis

Hypothesis Space :—

- The space of all hypothesis that 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 the machine is a different hypothesis about the f^n that maps input vectors to output vectors.

Exam (x, Y)

Training data ÷ set of examples

feature space — X

Concept — c, $c \subseteq X$

Target function f, $f : X \to Y$

Concern about choosing H.S

i) size
ii) Randomness
iii) Parameter

Inductive learning/prediction :—

- Need to make assumptions.
- experience alone doesn't allow us to make Conclusion about unseen data instances.

Two types of Bais—

- Restriction — limit the HypothesisSpace. (Polynomials — specifying the form of the f
- 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 exm.
- A Hypothesis is Consistent if it agrees with all training examples.
- A Hypothesis said to be ~~recognized~~ generalize well if it correctly predicts the value of y for novel example.

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.

Types of Inductive Bais —

- Occam's Razor :— the simplest consistent Hypothesis about the target fn is actually the best.
- Minimum description length :—
- Maximum Margin : When drawing the boundary b/w two classes, attempt to maximize the width of the boundary (SVM).

Generalization :—

Components of Generalization Error—

- Bias — how much the average model over all training sets differ from the true model.
  - Error due to inaccurate assumption/simplifications made by the model.
- Variance — how much models estimated from different training sets differ from each other.

Underfitting and Overfitting :—

Underfitting → model is too 'simple' to represent all the relevant class characteristics.

- High bias and low variance.
- High training error and ~~High~~ low test error.

Overfitting → Model is too 'complex' and fits irrelevant characteristics in data.

- low bias and high variance.
- low training error and High test error.

Evaluation :—→

H - Hypothesis space
S - Training data — $h \in H$
h - Learning algorithm.

Experimental Evaluation → Matrix Values
Error Matrix
Accuracy
Precision / Recall.

Evaluating Predictions :—

Suppose we want to make a prediction of a value for a target feature on example
- y is the observed value of target feature on example x.
- $\hat{y}$ is the predicted value of target feature on example x.

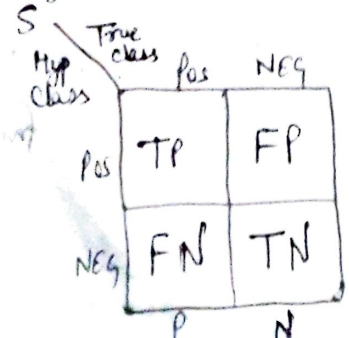
o How is the error measured ?

Absolute Error :— $|h(x) - y|$ for n training exm — $\frac{1}{n}\sum_{i=1}^{n}|h(x) - y|$

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

→ Linear Regression.

Classification → number of Misclassifications → $\frac{1}{n}\cdot\sum_{i=1}^{n}\delta(h(x), y)$



True Positive Rate

Accuracy = $\frac{TP+TN}{P+N}$

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

Recall = $\frac{TP}{P}$

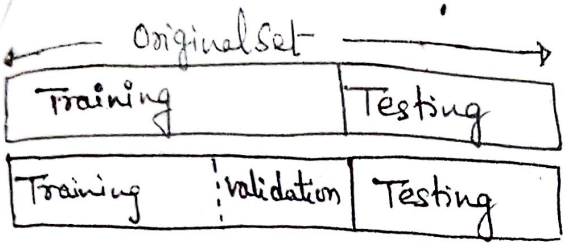
False Positive Rate

Accuracy = $\frac{FP+FN}{P+N}$

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

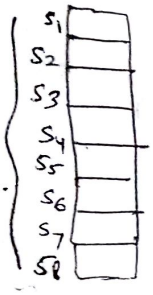
Recall = $\frac{FP}{P}$

Validation Set



Validation set is used during tuning training data to tune the model parameters check for the accuracy of your Hypothesis on the test set. So, when you do validation if your splitting data into these parts validation fails to use all the above data. to overcome this we use Cross validation.

K-fold Cross Validation :—



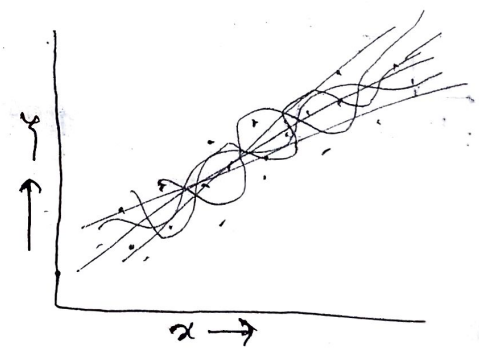
Round i : use S_i for testing
$S - S_i$ for training

	Round1	Round 2	Round 3	----	Round8
validation Accuracy →	93%	90%	91%	—	95%

Final Accuracy = Average (Round1, Round2, ----)

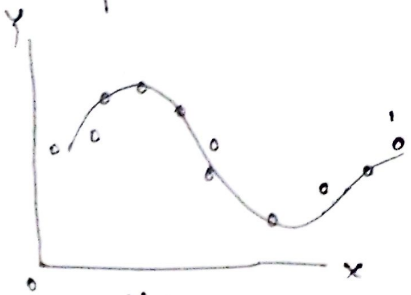
Linear Regression :—

Instance x, y
Given x predicts y
Simple Regressions → Single feature x.

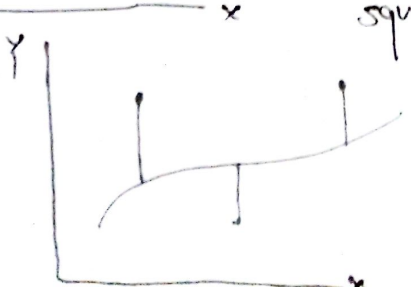


Example →

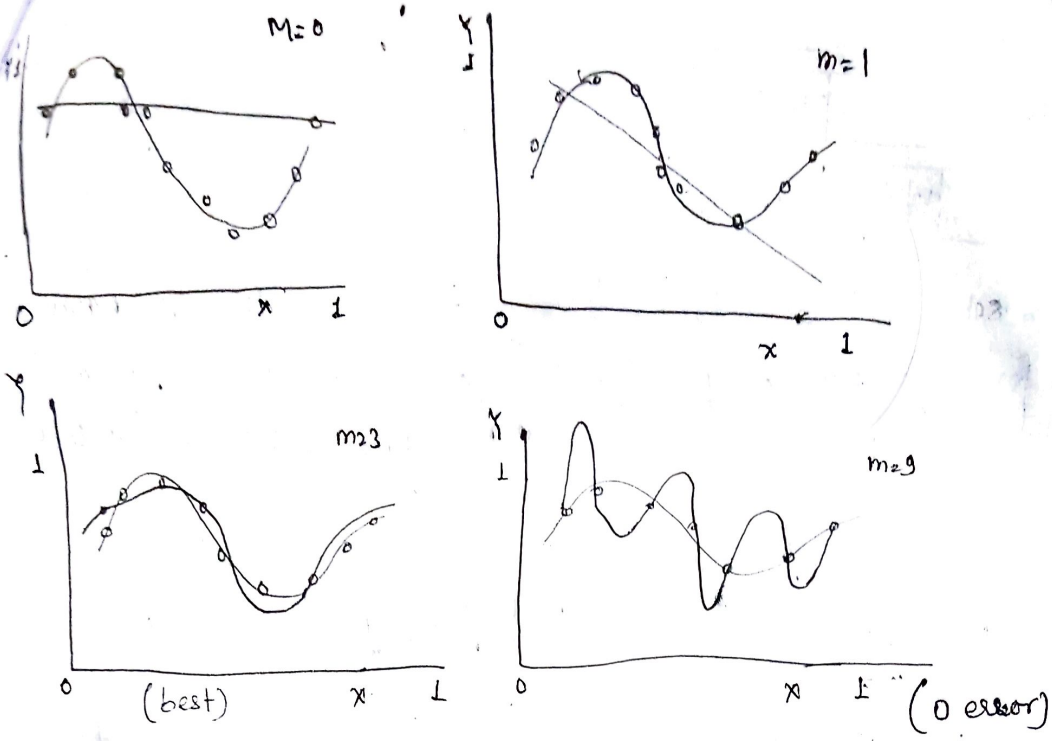
x & y both are Continuous



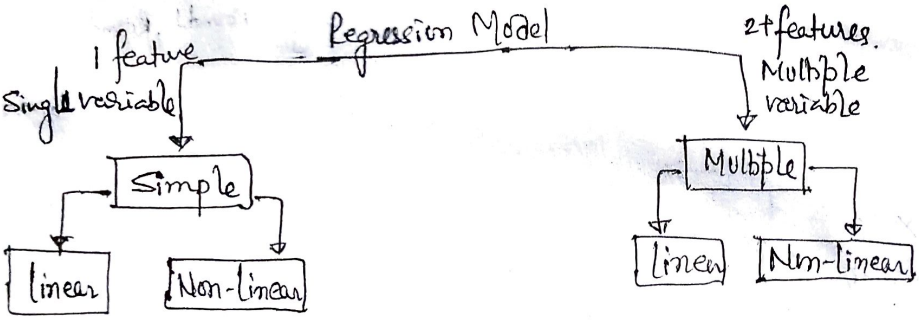
- The curve is true function (which is not given)
- We may use a loss function that measures the square error in the prediction of y for x.



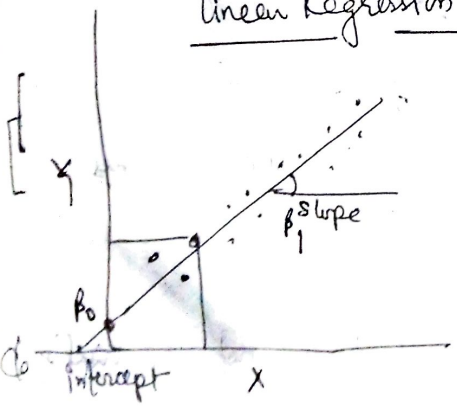
m(degree polynomial).



Sum of Square error is decreasing.



Linear Regression:—



$$Y = \beta_0 + \beta_1 x + \epsilon$$

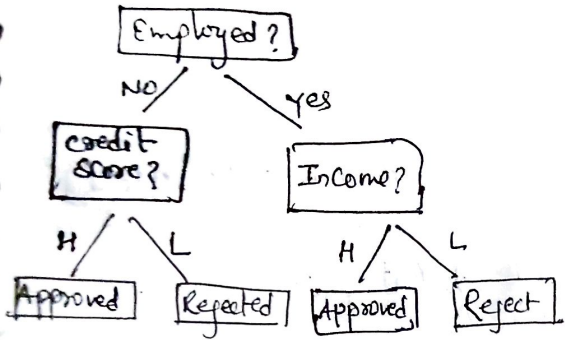
(β_0: point where it meet to Y-axis (Y-intercept); β_1: Slope of line; ϵ: error)

Decision Tree :→ It is a tree structured classifier, and it has two types of nodes, decision nodes and leaf nodes.

Decision Node, they specify a choice or a test based on this you can decide which direction you can go.

leaf node indicates the classification of example or the value of example.

* Decision Trees can be used both for classification and Regression. However, it is more popularly used for classification though they can be used for regression.



prefer Smaller Trees
— less depth
— Small no. of nodes.

Top down Induction of Decision Trees ID3 →

1. A ← the 'best' decision attribute for next node.
2. Assign A as decision attribute for node.
3. for each value of A create new descendant
4. Sort training examples to leaf node according to the attribute value of the branch
5. If all training examples are perfectly classified stop, else iterate over new leaf nodes.

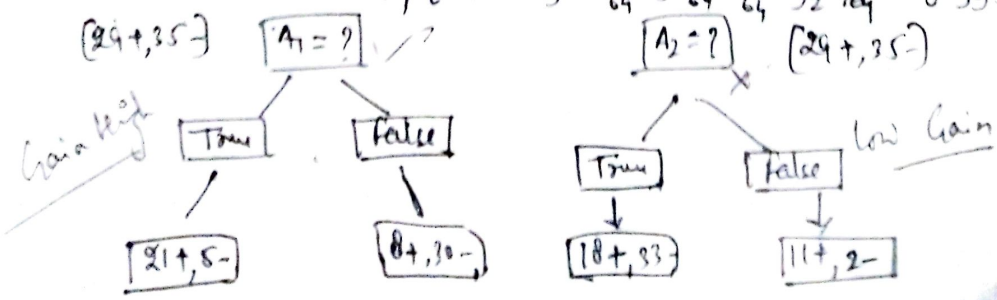
Choices

When to stop
- No more input feature.
- all examples are classified the same
- too few examples to make an informative split.

which test to split on.
- split gives smaller error.
- With multi-valued features.
  - split on all values or
  - split values into half

Exam

$$\text{Entropy}[29+,35-] = -\frac{29}{64}\log_2\frac{29}{64} - \frac{35}{64}\log_2\frac{35}{64} = 0.99$$

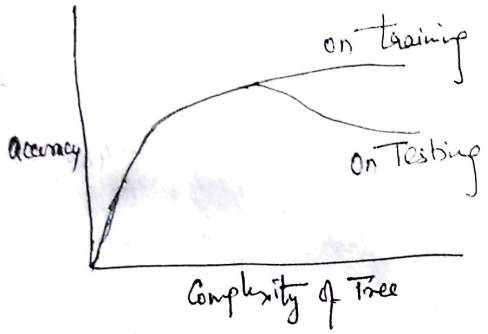


Gini Index —

$$Gini(Node) = 1 - \sum_{C \in classes} [P(C)]^2$$

$$GINI_{split}(A) = \sum_{v \in value(A)} \frac{|S_v|}{|S|} GINI(N_v)$$

Overfitting :— A Hypothesis 'h' is said to overfit the training data if there is another Hypothesis 'h'' such that h has more error than h on training data but h' has less error than h on test data.



Avoid overfitting →

→ Prepruning — Stop growing when data split not statistically significant.

→ Post pruning — Grow full tree then remove nodes.

K-nearest Neighbour :—

Training Method :— Save the training example.

At Prediction Time :— Find the 'k' training examples $(x_1, y_1) \cdots (x_k, y_k)$ that are closest to the test example x.

→ Classification → predict the most frequent class among those y_i's.

→ Regression → Predict the average of among the y_i's.

Algorithm →

Input :
D — Training data
K — Number of Neighbors
t — Input tuple to Classify

Output :
c — class to which t is assigned.

KNN Algo :—

N = 0 'N → set of Neighbours fi
for each d ∈ D do
if |N| < k then N = N ∪ d;
else if ∃ u ∈ N such that sim(t, u
then begin
N = N − u;
N = N ∪ d;
end.

...der distance formula:—

$x_i = (x_{i1}, x_{i2}, \ldots x_{iN})$

$x_j = (x_{j1}, x_{j2}, \ldots x_{jN})$

Disadvantage →

- Sensitive to k
- It is hard to determine good value of k
- However k = Sqrt (No. of training ...

dist-Euclidian (x_i, x_j) = $\sqrt{\sum_{m}^{N} (x_{im} - x_{jm})^2}$

Manhattan

$\sum_{i=1}^{n} |x_i - y_i|$

Representing Hypothesis →

most General Hypothesis — every attribute is positive example <?, ?, ?, ?>

most specific Hypothesis — No attribute is positive example <ϕ, ϕ, ϕ, ϕ>

General to Specific Ordering of Hypothesis —

h_1 = <Sunny, ?, ?, Strong, ?>

h_2 = <Sunny, ?, ?, ?, ?> ⇒ h_2 is more general than h_1.

→ First, for any instance x in X and hypothesis h in H we say that x satisfies h iff h(x) = 1.

→ Given Hypothesis h_j and h_k.

→ h_j is more general than or equal to h_k iff any instance that satisfies h_k also satisfies h_j.

Find-S : Finding A Maximally Specific Hypothesis —

1. Initialize h to the most specific Hypothesis in H.
2. for each positive training instance x.
   for each attribute 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.

Bayesian Learning

Bayesian probability — Partial belief.

Bayesian Estimation — Calculate validity of proposition.
- Prior estimate.
- New relevant evidence

Bayes Theorem →

$$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$$

- $P(h|D)$ is called posterior probability
- $P(h)$ is called Prior probability
- $P(D)$ - probability of data D.
- $P(D|h)$ is Conditional probability

Q →

$P(\text{Cancer}) = 0.008$, $P(-\text{cancer}) = 0.992$

$P(+|\text{Cancer}) = 0.98$, $P(-|\text{Cancer}) = 0.02$

$P(-|-\text{Cancer}) = 0.97$, $P(+|-\text{Cancer}) = 0.03$.

$$P(\text{Cancer}|+) = \frac{P(+|\text{Cancer})P(\text{Cancer})}{P(+)}$$

$$= \frac{0.98 \times 0.008}{P(+)}$$

$$P(-\text{cancer}|+) = \frac{P(+|-\text{Cancer})P(-\text{cancer})}{P(+)}$$

$$= \frac{0.03 \times 0.992}{P(+)}$$

Maximum A posteriori (MAP) Hypothesis —

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

$$h_{map} = \arg\max_{h \in H} P(h|D)$$

$$= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D|h)P(h)$$

Q →

mach 1 = 30 items/hr

mach 2 = 20 items/hr

Out of all produced parts — 1% are defective

out of all defective parts —
- 50% Came from mach 1.
- 50% Came from machine 2

What is prob that the part produced by mach 2 is defective?

P(mach 1) = 0.6 ✗
P(mach 2) = 0.4
P(defect) = 1%

P(mach1 | defect) = 50% ✗
P(mach2 | Defect) = 50%.

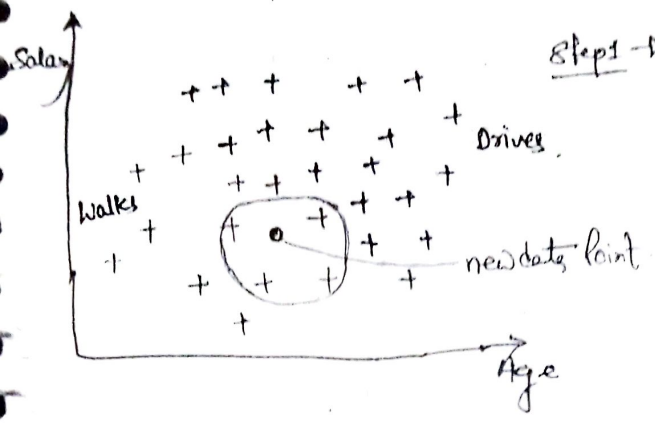
Now, we have to find out

$P(\text{defect} \mid \text{mach2}) = ?$

$$P(\text{defect} \mid \text{mach2}) = \frac{P(\text{mach2} \mid \text{defect})\, P(\text{defect})}{P(\text{mach2})}$$

$$= \frac{0.5 * 0.01}{0.4} = 0.0125 = 1.25\%$$

Naive Bayes :→



④ Posterior Probability ③ Likelihood ① Prior Probability

Step1 → $P(\text{walks} \mid O) = \frac{P(O \mid \text{walks}) * P(\text{walks})}{P(O)}$ ← ② marginal likelihood

Step2 →

$$P(\text{Drives} \mid O) = \frac{P(O \mid \text{Drives}) * P(\text{Drives})}{P(O)}.$$

Step3

$P(\text{walks} \mid x)$ vs $P(\text{Drives} \mid x)$

$$P(\text{walks}) = \frac{\text{Number of walkers}}{\text{Total No. of observations}}$$

$$P(\text{walks}) = \frac{10}{30}$$

$$P(O) = \frac{\text{No. of similar observation}}{\text{Total No. of observation}}$$

$$= \frac{4}{30}.$$

$$P(O \mid \text{walks}) = \frac{\text{No of similar observations among those who walks}}{\text{Total. No. of walkers.}}$$

$$2\ \frac{3}{10}.$$

$$P(\text{walks} \mid O) = \frac{\frac{10}{30} * \frac{3}{10}}{\frac{4}{30}} = 0.75$$

$$P(\text{Drive}) = \frac{20}{30}$$

$$P(O) = \frac{4}{30}$$

$$P(O \mid \text{Drives}) = \frac{1}{20}$$

$$P(\text{Drives} \mid O) = \frac{\frac{1}{20} \times \frac{20}{30}}{\frac{4}{30}} = 0.25.$$

$P(\text{walks} \mid O) > P(\text{Drives} \mid O)$.

New data Point should be walks to work.