MACHINE LEARNING in systems

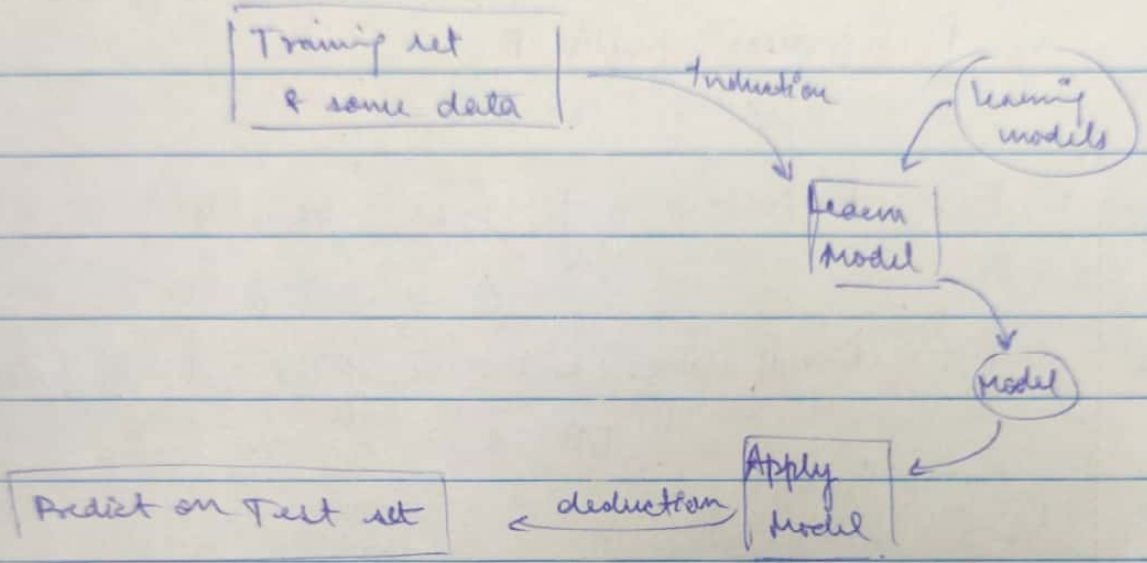
TM MITCHELL – A computer learns a task T from experience E, if its performance P improves with E.

E: Experience of past results.

T: Predicting out come of next or future
[TASKS]

P: Performance → Many metrics,
Accuracy, F1 score, R2 score, etc.,
probability.

Interaction of Data & Learning Algos.



It is primary divided into 3 types of learning

Supervised
- without ~~any kind~~ a of target to predict

Unsupervised
- without any kind of target of predict.

Re-inforcement
- If model does sm thg right, it is rewarded.
- If model does something wrong, it incurs a penalty.
- DEPENDS on training env.

SUPERVISED LEARNING → output is present.

↳ Regression → predict the value of a target feature which may have continuous values.

Eg predicting price.

↳ Classification → classify the data entry into one or more sets of classes.

Eg. classifying whether a person has breast cancer or not.

CROSS- VALIDATION of data

There may be a structure present in our data SUCH THAT train data DOES NOT CAPTURE the actual distribution of the full data.

What do we do ?

Cross validation - why? | To access the whole feature space.

↳ simply take different training sets & different test sets to eliminate bias.

|C1|C2|C3|- - -|C10|

first 6 & train, next 6 & train, next 6 & train,

- - - - - - ?

CLASSIFICATION : when the output var. is a category i.e. 'red' or 'blue'

REGRESSION : A regression problem is when the output is a [real] value. Eg 'dollars' or 'weights'

UNSUPERVISED ALGORITHMS

No target features to predict,

CLUSTERING: whenever we want to discover [GROUPINGS] in data

ASSOCIATION: An association rule learning is where you want to discover rules that describe large portions of data.

↓

Find an [Associate relationship] b/w some features to categorise.

SEMI - SUPERVISED & TRANSFER LEARNING

- SEMI - SUPERVISED → some target values are known but not for others.
- TRANSFER LEARNING → learning of one domain helped in learning of another.
  Eg movie recommender through novel likings.

Data, Info & Knowledge

- Data → raw facts & figures.
- Info → ordered data
- Knowledge → combination of facts & current info.

PROPERTIES of Data

1. VOLUME - quantity of the data you have.
2. VARIETY - The different forms of data.
   Eg. health care, images, video, audio.
3. VELOCITY - Freq / Rate of data generation.
4. VALUE - How much meaning the data adds to our model.
5. VERACITY - Certainty & correctness in data.

TRAIN, VALIDATE & TEST

- **TRAIN**: the data on which we want to build our model
- **VALIDATE**: the data which gives review while the training phase of model
- **TEST**: final set of data on which we evaluate our model.

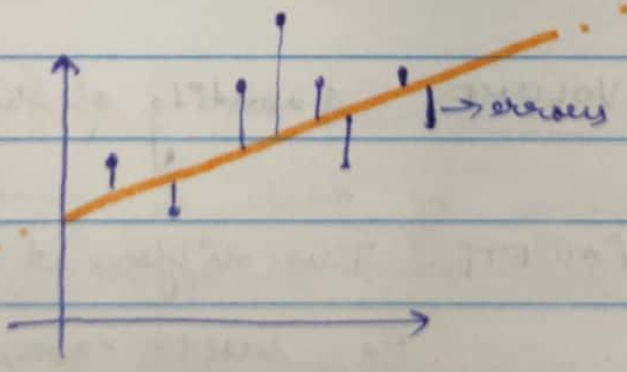
REGRESSION

We want to predict a continuous RESPONSE VARIABLE given some PREDICTOR VARIABLES.

→ Linear Regression

① * Predictive modelling technique to predict a CONTINUOUS response var.

② The independent vars. are predictors.



The relationship is thought of to be linear.

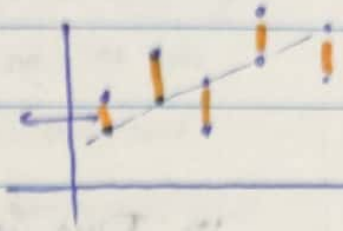
THE LEAST SQUARES METHOD

↳ $$m = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\Sigma(x-\bar{x})^2}$$

↳ This is the slope of the line which minimizes the SQUARED ERRORS.

Draw

MEAN SQUARED ERRORS



Actual values → y_i

Predicted values → from line → Y

$$E = \frac{\left[\Sigma (Y - y_i)^2\right]}{n}$$

this is a metric to evaluate how bad our values are wrt the ACTUAL VALUES.

R-SQUARED

→ It is a statistical measure of how good our model is predicting the response variable.

→ Eg $x, y \rightarrow 0.85\ R^2$

↓

85% of variability of y could be explained by x.

VARIABILITY: It tells us that the behaviour of y could be explained with x.

85% ~~varies~~ R^2 means → 85% of the time, this model can predict the response behaviour ~~varies~~ ~~values~~ correctly.

R^2 → Formula = $1 - \frac{\Sigma(y_p - y)^2}{\Sigma(\bar{y} - y)^2}$

(y_p → predicted, y → actual, $\bar{y}$ → mean)

$= 1 - \frac{RSS}{ASS}$

ASS → Avg. sum of squares.

RSS: Residual sum of squares → the sum of squares of the residues / errors) of the points wrt predicted values.

→ $\Sigma[(y_p - y)^2]$

ASS: ~~On avg., how does our model~~ The error in performance of an average model.

different methods to do LReg.

Linear Regression - Gradient Descent
+
Regularization.

Linear regression has 2 parameters →

$$Y = mX + c$$

m & c

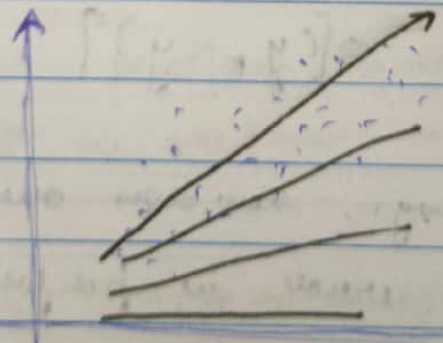
↳ we need to find m & c such that the (error) of our model is minimized.
↳ (RESIDUAL)

⇒ Gradient Descent

Slowly move in some direction.

which direction ?

The direction in which the error reduces the (MAX).



The values of m & c are updated until residual errors.

→ let us say residual error with the first step is E_1

→ If the next step gives us an error E_2, then we (update OUR LINE)

- We keep on moving our line to new parameters till we are getting a SMALLER ERROR.

Loss f^n in gradient descent

↳ The loss is the error in our predicted value of m & c.

↳ we need to minimize the value of THIS LOSS FN to get most accurate (m) & (c) values.

We will use - mean squared error.

MSE / RMSE → $\frac{\sum_i (y_i - y_{p,i})^2}{N}$

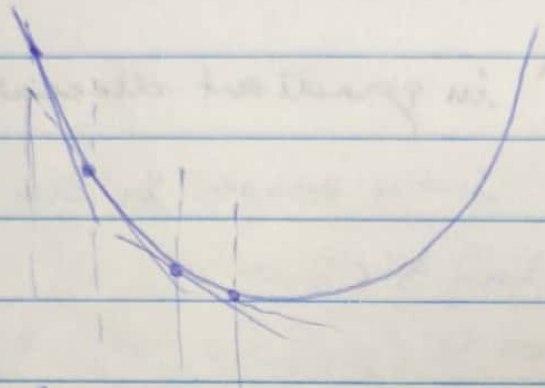
↓ this is our loss f^n.

$E = \frac{\sum_i (y_i - mx_i - c)^2}{N}$

VISUALIZE →.

→ steep slope: take large steps.

→ less slope & take small steps.



⇒ GD is an iterative algo. to find min. of a f^n.

→ Its like you get the slope at a point & if the slope is very high → move faster

→ If it is smaller slope you move slower.

$E \rightarrow f(m, c)$.

$$0 = \frac{\partial f(m,c)}{\partial m} \quad , \quad \frac{\partial f(m,c)}{\partial c} = 0$$

at the minima.

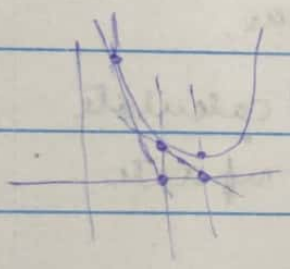
Applying gradient descent →

form $m = m$ & $c = c$,

$$\frac{\partial E}{\partial m} = \frac{1}{n} \sum_{i=0}^{n} 2(y_i - (mx_i + c))(-x_i)$$

$$D_m = \frac{2}{n} \sum_{i=0}^{n} (y_p - y_i) \cdot x_i$$

This gradient descent also has a LEARNING RATE, L - how big steps we take.

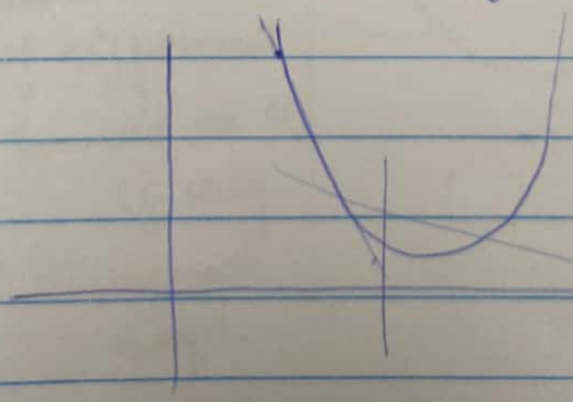


$$m_{new} = m_{old} - L \times D_m$$

$$c_{new} = c_{old} - L \times D_c$$

→ simultaneous updation

We go in the reverse direction of the slope of E wrt m & c.



→ large LRs mean that we may overshoot the minimum of our cost fn.

There are 3 types.

* BATCH GD

Take the whole dataset at each iteration.

↳ E is evaluated for all the points at each iteration.

* STOCHASTIC GD

↳ Here we are selecting |ONE ENTRY| at an ~~each~~ iteration & then do not consider that for future iterations.

↳ TAKE only 1 point, calculate error wrt that & update.

* MINI BATCH GD

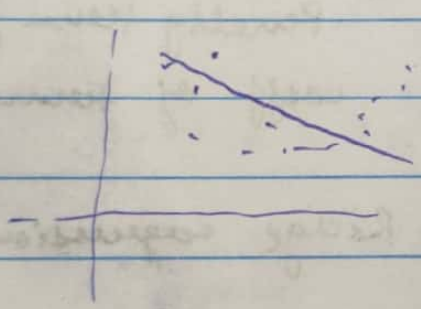
↳ Here, it uses n samples instead of 1, at each iteration.

REGULARIZATION –

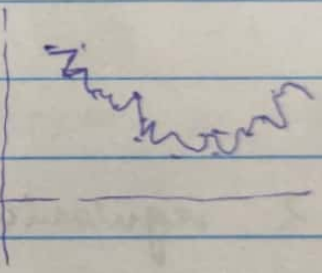
It is a technique for reducing model complexity & prevent over-fitting which may result from simple linear regression.

underfit & overfit are problems which regularization deals with.

UNDERFIT

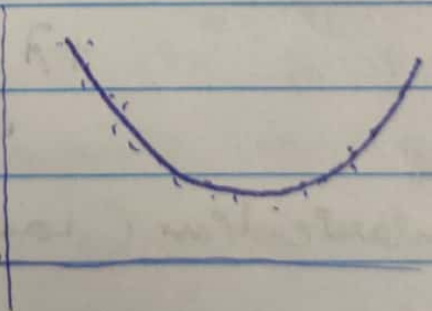


OVERFIT



↙ Through regularization

tries to make the contribution of different powers as 0, if they were not reqd.



Ex → Let us say we have

$$h_\theta(x_i) = \theta_0 x_a + \theta_1 x_b^2 + \theta_2 x_c^3$$

⇒ Then, the mse becomes -

$$mse = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - h\theta(x_i)\right)^2 + \lambda \underbrace{\sum_{i=1}^{n} \theta_i^2}$$

Penalty term for coeff. of terms.

L2 regularization (Ridge regression).

$$mse = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{j=1}^{n} \theta_i^2$$

L1 regularization (Lasso Regression)

$$mse = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - h_\theta(x_i)\right)^2 + \lambda \sum_{j=1}^{n} |\theta_j|$$

Okay, we will have →
partial derivatives wrt.
each parameter & then
we start updating.

Eg. $h(x) = c + m_1 x_1 + m_2 x_2 + m_3 x_3$

$N = 300$, items = 10

1. Initialize c, m_1, m_2, m_3 as 0.
2. SGD for $i = 190$

3. Δm & Δc → m_1, m_2, m_3 & c.

$\frac{\partial mse}{\partial m_1}$, $\frac{\partial mse}{\partial m_2}$, $\frac{\partial mse}{\partial m_3}$, $\frac{\partial mse}{\partial c}$

Put values of x_1, x_2, x_3 for
$i = 190$ in h & then
update m_1, m_2, m_3 & c
as

$m_1 = m_1 - L \times \Delta m_1$

$m_2 = m_2 - L \times \Delta m_2$

⋮

$c = c - L \Delta c$

LOGISTIC REGRESSION

Ways to express probability :

$p(0_1) = p$

$\bar{p}(0_1) = 1 - p = q.$

	notation	values.		
Standard prob.	p	$p=0$ $q=1$	$p=0.5$ $q=0.5$	$p=1$ $q=0.$
odds	p/q	0	1	∞
log. prob.	$\log(p/q)$	$-\infty$	0	∞

LOG ODDS

→ if the prob. of selection is equal,

$\text{logit}(p/q) = 0.$, log odds → 0

→ if the odds in favour of one event is $+x$, $x = \log\left(\frac{p}{q}\right)$

then odds in disfavour of other event q are $-x$. $\rightarrow$ $-x = \log\left(\frac{q}{p}\right)$.

From prob. to log odds.

$$\boxed{z = \ln \frac{p}{1-p}} \rightarrow \text{logit fn.}$$

z - is a hypothesis

$$e^z = \frac{p}{1-p}$$

$$(1-p)e^z = p$$

$$e^z = (1+e^z)p$$

$$p = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

$$\boxed{p = \frac{1}{1+e^{-z}}} \text{ logistic fn}$$

How to use as classification model?

↳ if for a value $> z_1$, $p = 0.7$

NOW, OUR BASIS

↳ $p \geq 0.7 \rightarrow 1$

↳ $p < 0.7 \rightarrow 0$.

↳ $z_1 < z \rightarrow p \geq 0.7, \rightarrow 1$.

↳ $z_1 > z \rightarrow p < 0.7, \rightarrow 0$.

↳ $\boxed{z > z_1, \quad 1}$

⇒ $\boxed{z < z_1, \quad 0}$.

Now, this model was a multi-dim space →

$$z = \alpha + \beta \cdot x$$

$$z = \alpha + \boxed{\beta_1 x_1 + \beta_2 x_2 + \dots \beta_k x_k}$$

↳ Now, Map z to the range 0 to 1 using logistic fn –

$$p = \frac{1}{1 + e^{-z}}$$

Now, let us say we have

A point $z_1 \rightarrow \begin{pmatrix} x_1 & x_2 & \equiv & x_3 \\ 3 & 4 & & 10 \end{pmatrix}$

↓

Here, we will have a decision boundary

or rather, any combo. which gives,

$x_1 \geq 3$

$x_2 \geq 4$

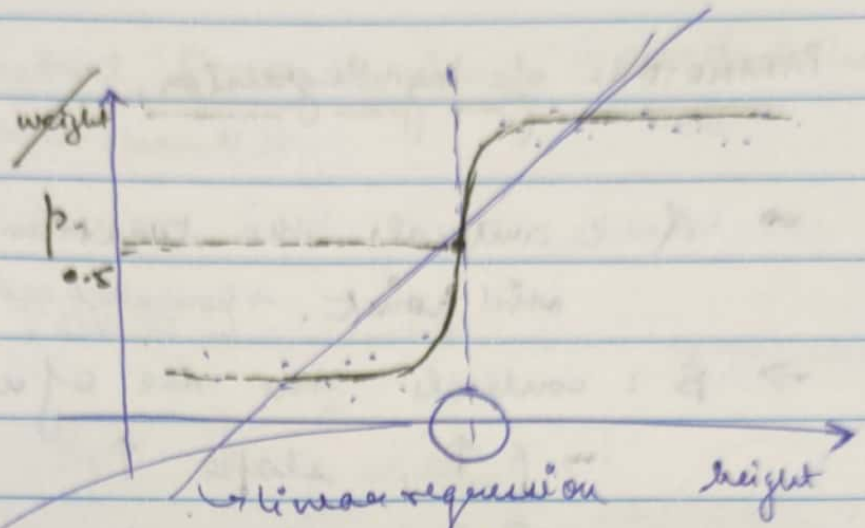
$x_3 \geq 10$

$\boxed{\text{If } (z) \geq 0.7}$ ↳ as 1

&

$\boxed{\text{If } (z) < 0.7}$ as 0.

Eg:



↳ linear regression has a problem

↳ NO CONSTRAINTS.

↳ In logistic regression, we just input the height.

↳ when $f(H)$ is > 0.5

↳ OBESE

else

↳ NON-OBESE.

this provides a decision boundary in feature space. → means

$z_0 = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$

- z_0 is boundary value.
- $p(z_0)$ is threshold.

Any combo of input vars such that z value is $> z_0$, classified as 1, else 0.

PARAMETERS of log Regression.

→ α : controls the location of mid point. → how much shift has to be there.

→ β : controls the rise of model.

↳ $\beta \uparrow$, slope ↑.

$\beta \downarrow$, slope ↓.

Adjusted R-SQUARED

↳ This is used to take into account the dimensionality of model too.

↳ Not necessary that the higher param model is better.

	R^2	adj. R^2
1 param	0.67	0.63
2 params	0.83	(0.82) → maybe the best
3 params	0.88	0.7

⑧ ASSUMPTIONS for LINEAR MODELS to work.

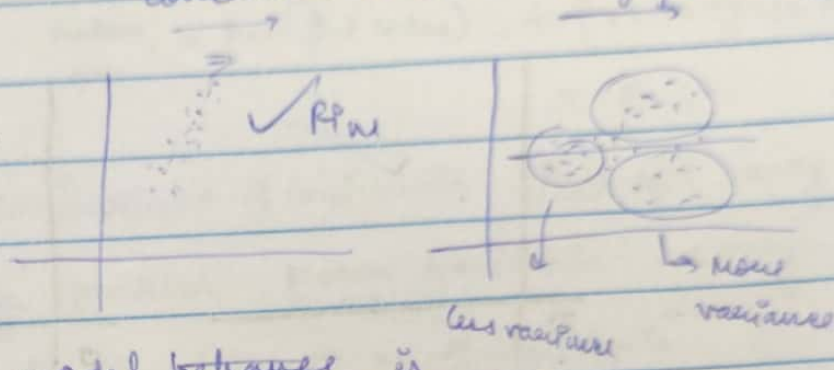
⊕ There | terms must hold true for linear model to WORK correctly

↓

Constant Error Variance — Needs to hold true

↳ Homoskedasticity

↳ Variance in the errors should be similar.



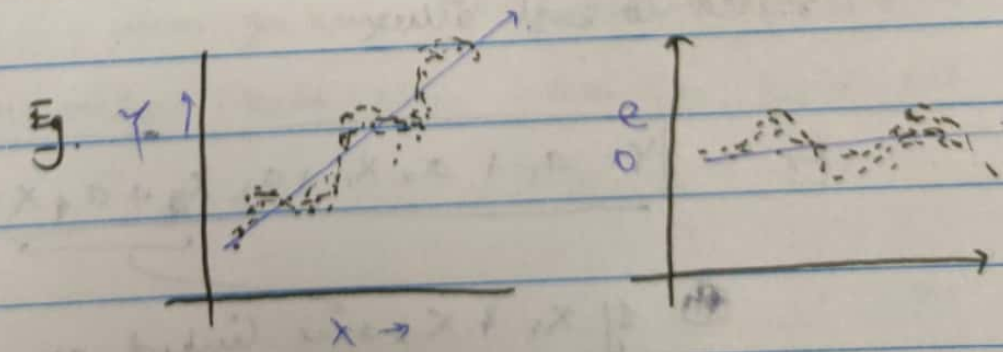
The model ~~behaves~~ is fit correctly in ↳ First CASE

But NOT in Second

Independent Error Terms. — Needs to hold true.

Each point must have an independent error (not dependent on any other points).

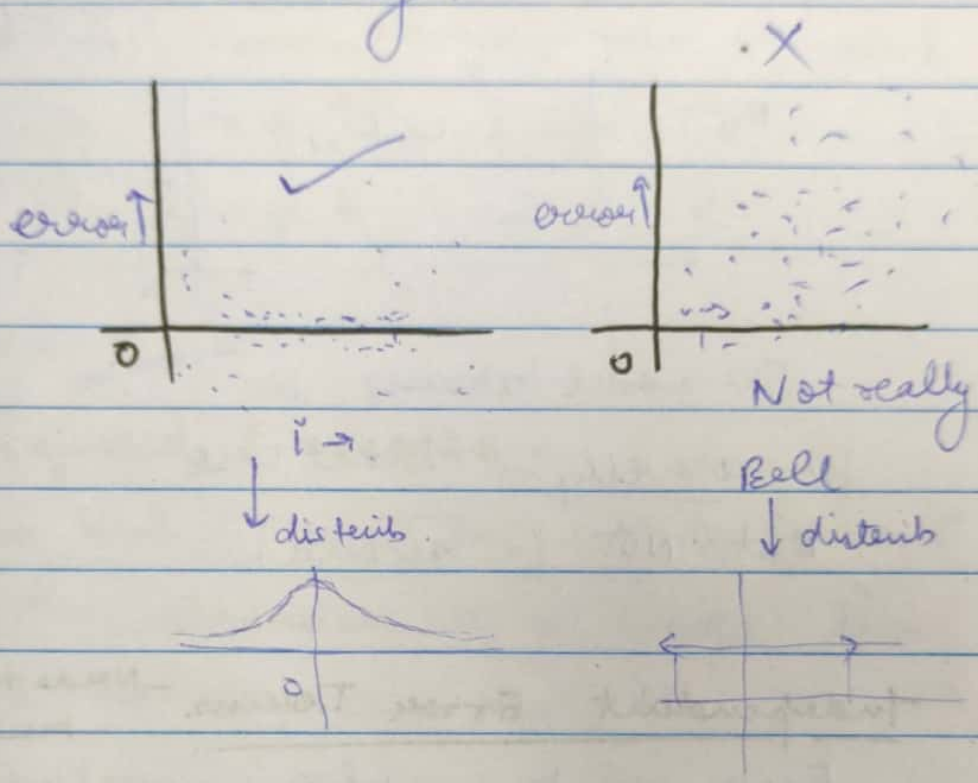
↳ Errors must be UN-CORRELATED.



Linear model does not work very well.

Normality of errors :

Most of the points must lie around the best fit line for the model to work nicely.



No multi- collinearity -

Multi collinearity occurs when X's are related to each other.

Ex $Y = a_1 + a_2 X_1 + a_3 X_2 + a_4 X_3$

If X_2 & X_3 are linked or correlated, then if we include only 1, then it is going to be redundant.

↳ We try to avoid collinearity amongst vars,
* only keep one of 2 vars if they are highly related.

Eg. Motor accidents = $\beta_0 + \beta_1$(cases) $+ \beta_2$(no. of residents) $+ \varepsilon$

* If cases $\propto$ no. of residents, then we may only predict Motor Accidents ~~as~~ only by cases or no. of residents.

↳ we can ~~only~~ also try to remove the variable & then try to see how model behaves.

↳ Variance Inflation Factors are used for removing features.

NEW THOUGHT: A feature inclusion would matter when you have 100 features & a billion records. Data size changes by a lot.

EXOGENITY : Dependence on unknown features.

There may be other features or biases affecting the dependent variable.

when these features are omitted, model doesn't behave very well.

Eg salary$_i$ = $c_0 + c_1$(years of edn)$_i$ + ε_i

↳ This is not a very good inference.

↳ We may know that the years of edn affects the salary but it may not be a full CAUSE for the salary.

ADJUSTED R^2 → detail.

↳ It penalizes (rightly) for adding features WHICH DO NOT HELP in PREDICTING the dependent variable.

↳ The formula for adjusted R^2 is =>

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$\frac{SS_{res}}{(df_e)}$ → deg. of freedom $(n - (p) - 1)$
↳ p removed features.

$\frac{SS_{tot}}{(df_t)}$ → ↳ $(n-1)$ → total deg. of freedom.

Eg: Features = 10

Removed 4, $p = 4$

$df_e = 10 - 4 - 1$

$\boxed{df_e = 5}$

$\boxed{df_t = 9}$

using R^2 itself for $R^2_{adjusted}$ →

$R^2_{adjusted}$ $1 - \underline{(R^2 - 1) \ldots}$
$\ldots$

write...

Challenges in ML

→ AVOID OVER-LEARNING.

① Hold Out : keep some test data & don't even show test data to model.

② Cross validation :

③ Data Augmentation : to handle class imbalance.

④ Curse of Dimensionality

⑤ L1, L2 regularization.

⑥ Feature Selection & Reduction :

- We just select out from the set
  A_1 A_2 A_3 A_4
  ↓
  take these.
- Here, we COMBINE variables & transform vars.
  → A_1 A_2 A_3 A_4
  ↳ $A_1 \times A_2$ | $A_3 \times A_4$

→ MODEL EVALUATION.

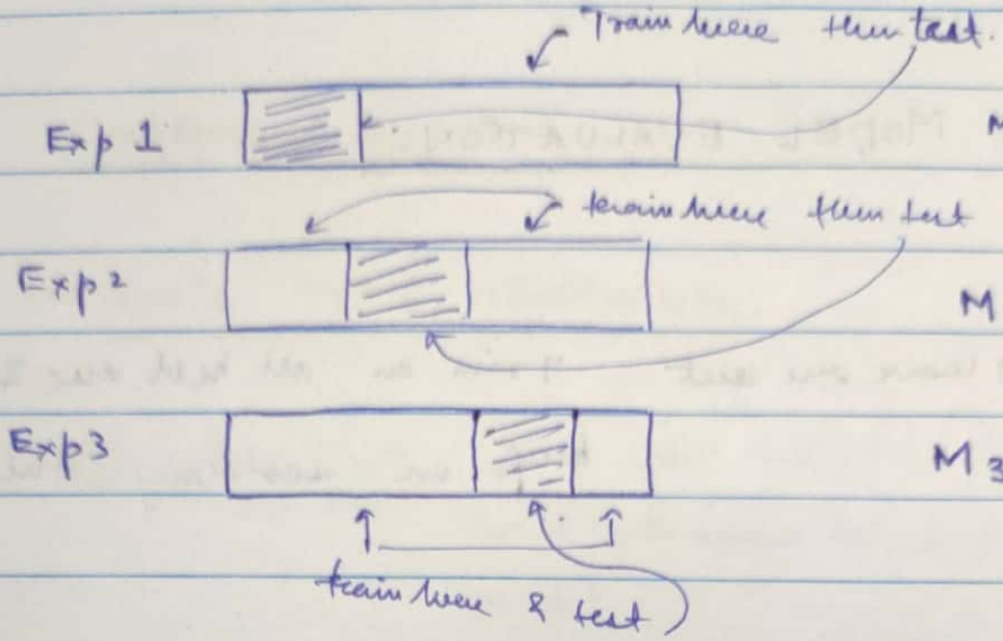
↳ Leave one out : Train on all but one & keep on doing this.

↳ Hold out : Divide dataset into 2 sets
train & test sets
↳ validation
learning
: $2/3^{rd}$ train & unseen $1/3^{rd}$ test

: Repeated Holdout.

→ k- Fold Cross validation :

↳ Divide data into k equal parts.
↳ ONE for test.
↳ OTHER for training.
↳ Process is repeated for all folds & accuracy is averaged.



BOOTSTRAP SAMPLING.

↳ Cross validation uses WITHOUT REPLACEMENT sampling.

→ Bootstrap sampling is WITH REPLACEMENT to form the training set.

↳ n size : TRAIN set.

↳ Take out n instances with replacement

→ Use this data as training.

→ USE instances from original which are NOT IN NEW TRAIN SET.

Evaluation Metrics

↳ **ACCURACY** → $\frac{\text{correct classified}}{\text{total no. of instances}} \times 100$.

↳ assumes equal cost for each of the classes.

↳ what does that mean ?

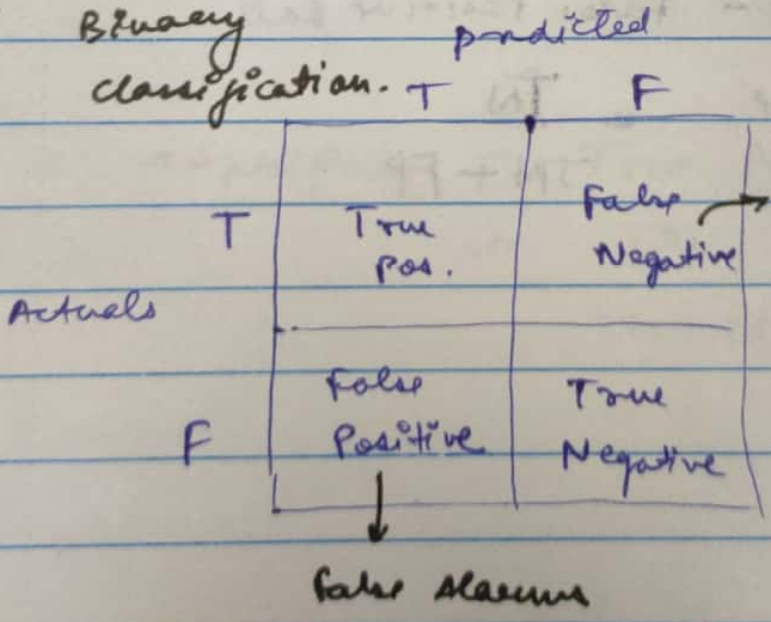
→ 1000 total → 990 healthy, 10 unhealthy

↳ 99% accurate if we just predict healthy.

↳ This is wrong...

↳ CONFUSION MATRIX & F1-SCORE

Binary classification.



True Positive Rate : $\frac{TP}{TP + FN}$ = TPR.

why? because FN↑, TPR should ↓

False positive Rate : $\frac{FP}{FP + TN}$

why? because TN↑, time nada ~~salah~~ negative mean lawan False Positive.

SENSITIVITY

↳ $\frac{TP}{TP + FN}$ = How correctly we detect positive classes.

True positive detection

SPECIFICITY

↳ 1 - False Positive rate

↳ $= \frac{TN}{TN + FP}$

True Negative detection

		Pos	Neg. ← Predicted
Q ⇒	Pos	22	0
Actuals →	Neg	17	0.

True Pos = 22

True Neg = 0

False Pos = 17

False Neg = 0

→ Accuracy = $\frac{22}{22+0+17+0} = \frac{22}{39}$

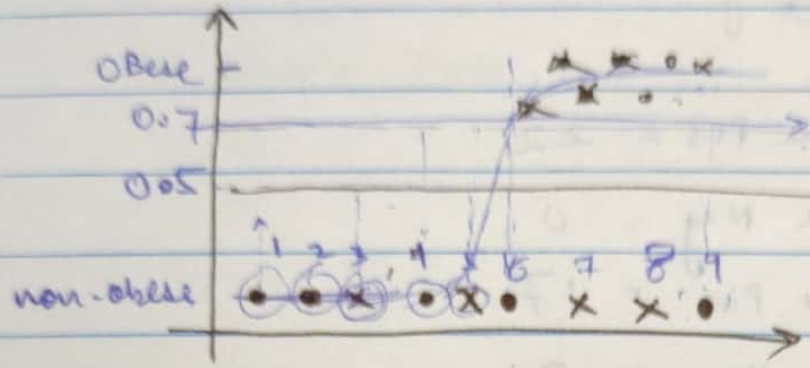
→ Sensitivity → True Positive Rate

⇒ $\frac{TP}{TP+FN} = \frac{22}{22} = 1.$

→ Specificity → True Negative Rate

⇒ $\frac{TN}{TN+FP} = \frac{0}{0+17} = 0$

Logistic Regression : cont.



When we have a threshold of 0.5 this is the particular graph.

Case 1 → 0.5 threshold - how does it effect?

Actual \ Predicted	0	1
0	3 (1,2,4)	2 (6,9) → were non-obese classified as obese.
1	2 (3,5)	2 (7,8)

0 → Non-Obese

1 - Obese

Case 2 → 0.7 threshold - how does it effect.

Actual \ Pred	0	1
0	4 (1,2,4,6)	1 (9)
1	2 (3,5)	2 (7,8)

→ CONCLUSION

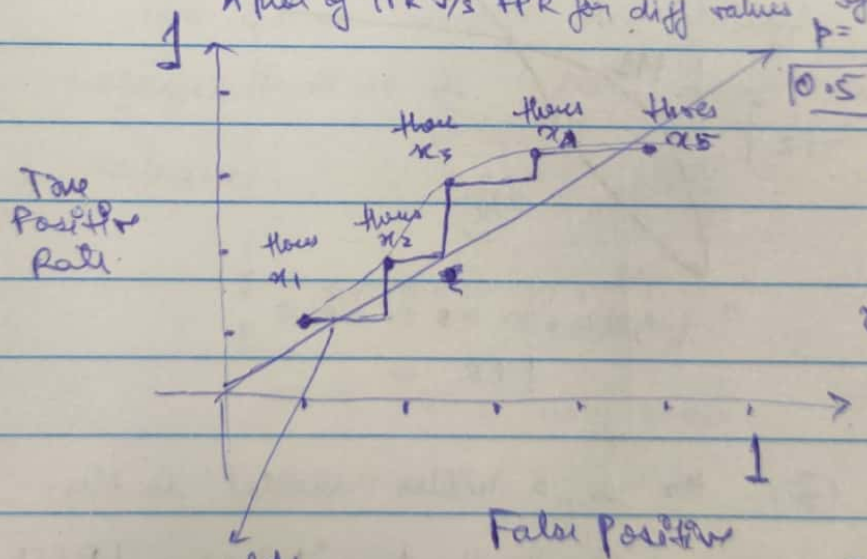
For the same model, if I CHANGE THE THRESHOLDS, the CONFUSION MATRIX changes.

→ This is nice !!

(#) How to do this effectively, instead of changing threshold & calculating confusion matrix again & again, PLOT !

ROC curves → Reciever Op^n Characteristics

A plot of TPR v/s FPR for diff values of threshold values

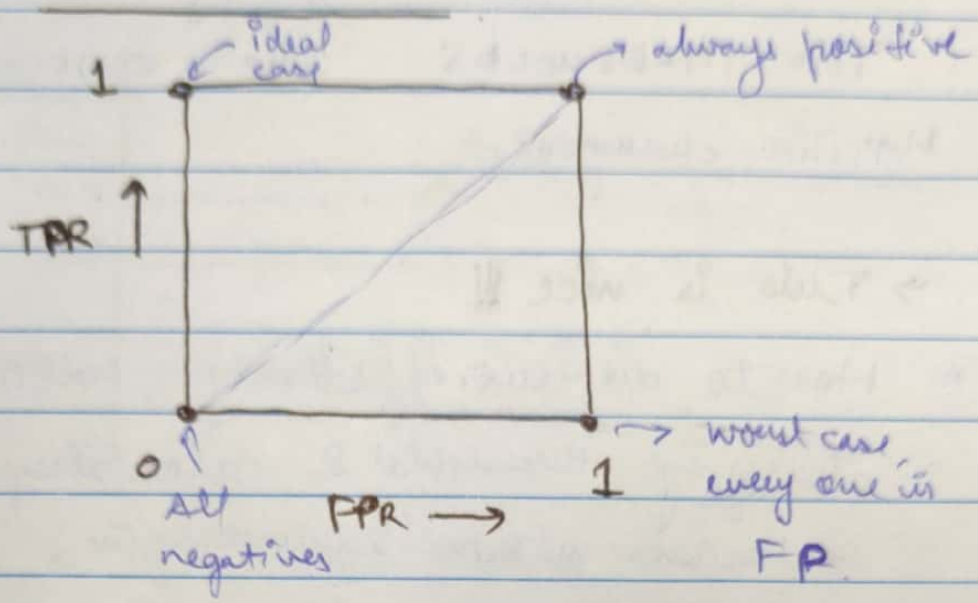


p = 0.5 → when ratio is 1, means random model

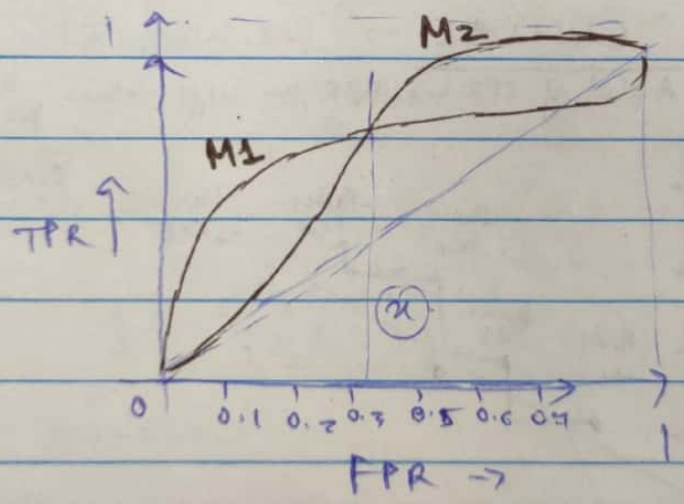
(#) Thresholds need not be related.

(#) Each & every point is a model evaluation against a p value threshold p.

Roc curve chart.



Roc curve evaluation.



(#) M2 is a better model in the sense that it has a LESSER FPR after x point for a greater TPR.

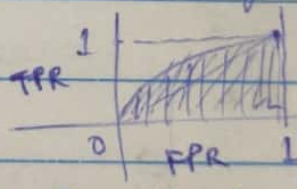
(A) If we want a very high TPR & low FPR, then M_1 would be better as M_1 would although not give a very high TPR (BUT) (FPR) is very low.

less False Alarm → M_1.

Better TPR & → M_2.
(regardless of what is FPR)

AREA under the ROC curve

The area under the graph is representative of how good the model behaves.



0.9 - 1 - Excellent

0.8 → 0.9 - very good.

0.7 → 0.8 → good

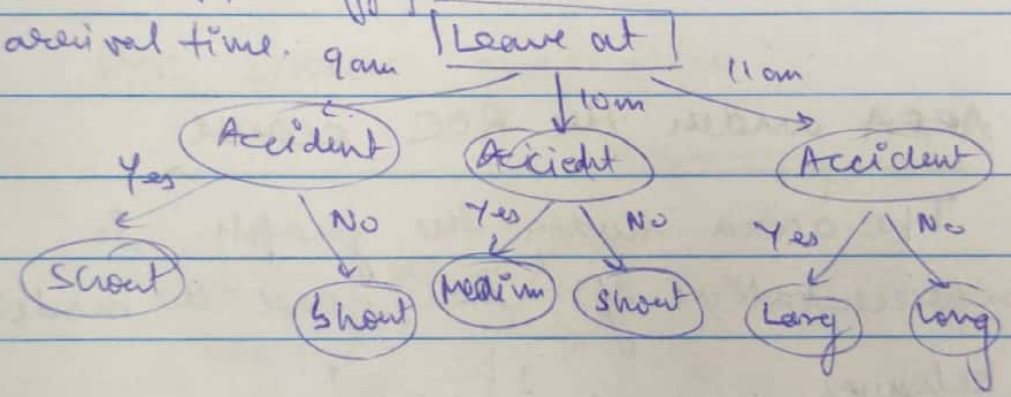
< 0.7 okayish..

DECISION TREES

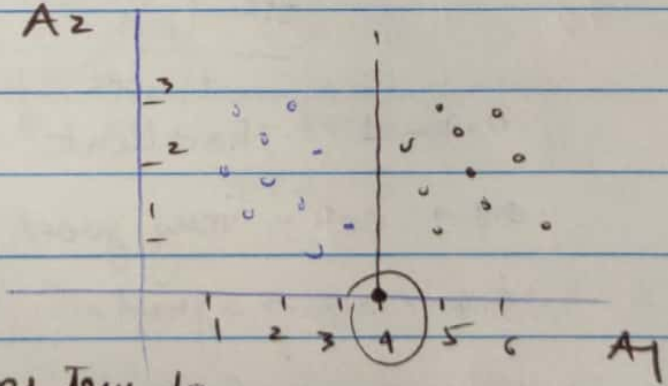
① It is an inductive learning task
↳ we use a dataset to find a SPLIT POINT.

② Each non-leaf node represents a DECISION.

Eg.
Predicting / Classifying arrival time.

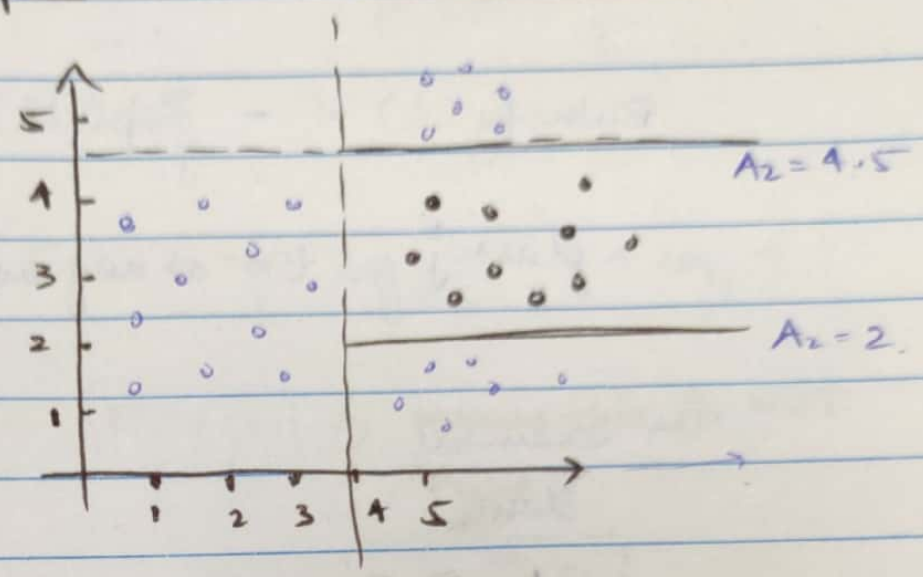


Eg.



Decision Trees try to find the SPLIT POINTS $A_1 = 4$ is a DECISION BOUNDARY.

Examples



~~Dec Tree~~ How

Decision Trees try to find regions of Homogeneity of data.

How is homogeneity defined?

↳ A measure is ENTROPY of data.

ENTROPY is the metric which is used for finding out split points.

↳ The attrib. value which gives the min. entropy is taken as a DECISION BOUNDARY.

ENTROPY

$$\text{Entropy}(t) = -\sum_j p(j|t) \log_2 p(j|t)$$

class j, $t \rightarrow$ at any node t.

For example.

classes.

c_1	$\Rightarrow 0$
c_2	$\Rightarrow 6$

How, feature set does not matter.

(1) $\hookrightarrow$ $E(s) = -\sum_j p(j|t) \log_2 p(j|t)$

$p(c_1) = 0$ for all,

$p(c_2) = 1$ for all.

$E(s) = -(0 \times \log_2(0) + 1 \log_2(1))$

$\boxed{E(s) = 0}$

$C_1 = 2$ $\quad p(C_1) = \frac{2}{6}$

(2) $C_2 = 4$ $\quad p(C_2) = \frac{4}{6}$

Entropy - ?

$$E(S) = -\left[2 \log_2 \frac{2}{6} + 1 \log_2 \frac{4}{6}\right]$$

$E(S) \neq 0$ → Means this is NOT Pure.

EXAMPLE ⟹

Let us say we have a data set →.

Day	outlook	Temp	Humid	wind	Play
	Sunny, overcast, Rain	Hot, mild, cool	High Normal	weak strong	Yes/No

How do we determine split point?

For each attribute, we can calculate the entropy value & GAIN

$$GAIN(S, A) = Entropy(S) - \sum \left[\frac{|S_v|}{|S|}\right] \times \text{entropy (value of } A_v)$$

GAIN → At an attribute how much do we gain.

S → whole data

A → Attribute

Eg. Gain (S, out)

= Entropy (S) — sum of all entropies for the attrib out.

G (S, out) = Entropy (S) - [Entropy (sunny) + Entropy (overcast) + . . .]

E(S) is just the ENTROPY of data set (class values)

(1) $E(S) = -\left[\frac{9}{14}\log\left(\frac{9}{14}\right) + \frac{5}{14}\log\left(\frac{5}{14}\right)\right]$

$E(S) = 0.939$

(2) Entropies for each attribute value.

OUTLOOK E(sunny) + E(overcast) + E(Rain)

How to calculate ↑ ?

For each attribute value calc. the no. of yeses & nos.

	Yes	No	Total.
sunny	2	3	5
overcast	4	0	4
Rain	3	2	5

how many data rows have sunny outlook & play as yes.

Now, entropy for sunny →

$$E(\text{sunny}) = \frac{5}{14}\left[\frac{2}{5}\left(\log\frac{2}{5}\right) + \frac{3}{5}\log\left(\frac{3}{5}\right)\right]$$

(Yes: 2, No: 3)

5 because we are considering the case in which the outlook is sunny.

$$E(\text{overcast}) = \frac{4}{14}\left[\frac{4}{4}\log\left(\frac{4}{4}\right) + 0\right]$$

(Yes: 4, No: 0)

$$E(\text{Rain}) = \frac{5}{14} \times \left[\frac{3}{5}\left(\log\left(\frac{3}{5}\right)\right) + \frac{2}{5}\log\left(\frac{2}{5}\right)\right]$$

(Yes: 3, No: 2)

TEMPERATURE

Gain(S, temp) = E(S) − [Entropy(hot) + Entropy(mild) + Entropy(cold)]

	Yes	No	Total
hot	2	2	4
mild	4	2	6
cold	3	1	4

$$E(\text{hot}) = \frac{4}{14}\left[\frac{2}{4}\log\left(\frac{2}{4}\right) + \frac{2}{4}\log\left(\frac{2}{4}\right)\right]$$

$$E(\text{mild}) = \frac{6}{14}\left[\frac{4}{6}\log\frac{4}{6} + \frac{2}{6}\log\frac{2}{6}\right]$$

$$E(\text{cold}) = \frac{4}{14}\left[\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right]$$

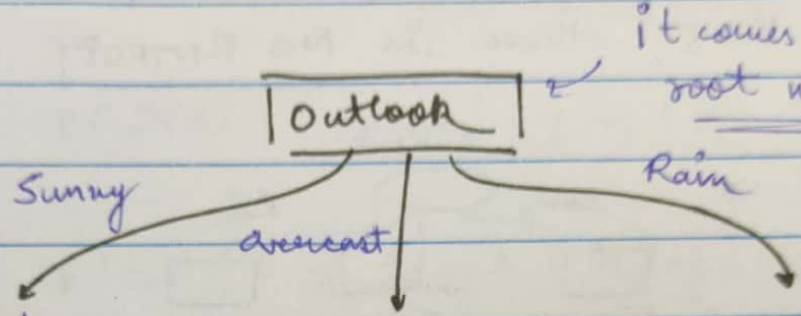
All of the above calculation is to find a split point at the root level!

IG(S, outlook) = 0.247 → highest IG.

IG(S, temp) = 0.029

IG (S, hum) = 0.152

IG(S, wind) = 0.048

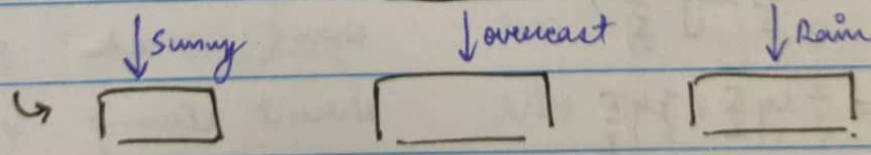


↳ 3 branches as we have 3 choices in the outlook variable.

This branch means we have all the elements having 'SUNNY' outlook.

Now, data set is reduced. How?

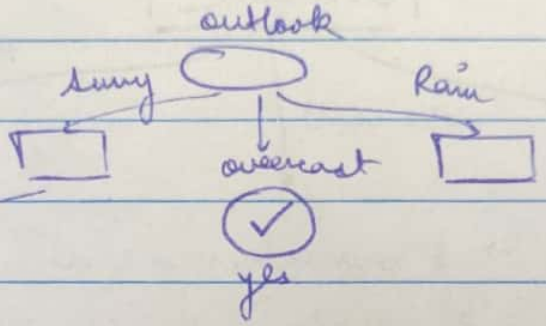
We take SUNNY outlook rows!



↳ these boxes are info. gain nodes FOR EACH Dataset pertaining to that node.

Some Inferences →

Since each & every row in the OVERCAST ROWS have target as YES, there is NO ENTROPY.



→ ① Info gain at sunny node.

Dataset [S']	Temp	Humidity	Wind	Play
Hot	High	Weak	No
Hot	High	Strong	No
Mild	High	Weak	No
cool	Normal	Weak	Yes
Mild	Normal	Strong	Yes

$E(S') = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right)$

$= \frac{2}{5}\log\frac{5}{2} + \frac{3}{5}\log\frac{5}{3}$

$= 0.2 \times 1.322 + 0.6 \times 0.737$

$= 0.971$

for this.

① IG (S', Temp) →

	Yes	No	Total.
Hot	0	2	2
Mild	1	1	2
Cool	1	0	1

$IG(S', T) = E(S') - [\text{Entropy (Hot)} + \text{Entropy (Mild)} + \text{Entropy (cold)}]$.

$E(\text{Hot}) = -\left[\frac{0}{2}\log\frac{0}{2} + \frac{2}{2}\log\frac{2}{2}\right] \times \frac{2}{5} = 0$

$E(\text{Mild}) = -\left[\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right] \times \frac{2}{5} + \left[\frac{2}{5}\right] = 0.4$

$E(\text{cool}) = -\left[\frac{1}{1}\log 1 + 0\right] \times \frac{1}{5} = 0$

$E(S', T) = 0.971 - 0.4$

$= 0.571$

② $IG(S'', \text{Humidity})$

S''	~~Temp~~ ~~Humid~~ ~~Wind~~ ~~Play~~	Yes	NO	T
High		0	3	3
Normal		2	0	2.

$E(\text{Humidity}) = -\left[\frac{3}{5} \times \left[0 + \frac{3}{3}\log 1\right]\right]$

0

$-\left[\frac{2}{5}\left[\frac{2}{2}\log 1 + 0\right]\right]$

$E(\text{Humid}) = 0$

③ E(wind)

	Y	N	T
weak	1	2	3
strong	1	1	2

$$E(\text{wind}) = -\frac{3}{5}\left[\frac{1}{3}\lg\frac{1}{3} + \frac{2}{2}\lg\frac{2}{3}\right]$$

$$-\frac{2}{5}\left[\frac{2}{2}\lg\frac{1}{2}\right]$$

$$= +\frac{3}{5}\left[\frac{1}{3}\lg 3 + \frac{2}{3}\lg\frac{3}{2}\right]$$

$$+\frac{2}{5}\lg_2 2$$

$$= 0.4 + 0.6 \times \left[\frac{1}{3}\lg 3 + \frac{2}{3}\lg 3 - \frac{2}{3}\lg 2\right]$$

$$= 0.4 + 0.6\left[\lg 3 - \frac{2}{3}\lg_{10} 2\right]$$

$$= 0.4 + 0.6\,[1.5849 - 0.6777]$$

$$= 0.4 + 0.6\,[\ldots]$$

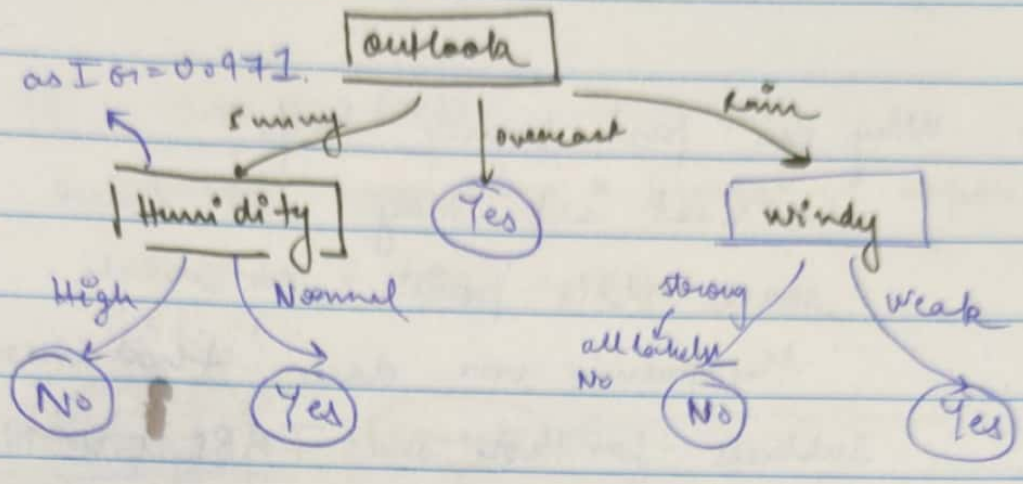
$$E(\text{wind}) = 0.94432$$

$IG(S', Temp) = 0.571.$

$IG(S', Humidity) = \boxed{0.971.}$ ✓

$IG(S', Wind) = 0.019$

as $IG = 0.971$.



* You do not need to calculate IG if you have a (PURE) branch

→ No noise.

What if we have very large no. of features?

⇒ It would be very computationally expensive.

PRUNING → If a branch or a root to leaf path or a subtree doesn't give any improvement (accuracy, no can remove.

↳ Pre-Pruning (early stopping).

↳ Try to stop at a certain DEPTH | No. of samples belonging to a LEAF NODE.

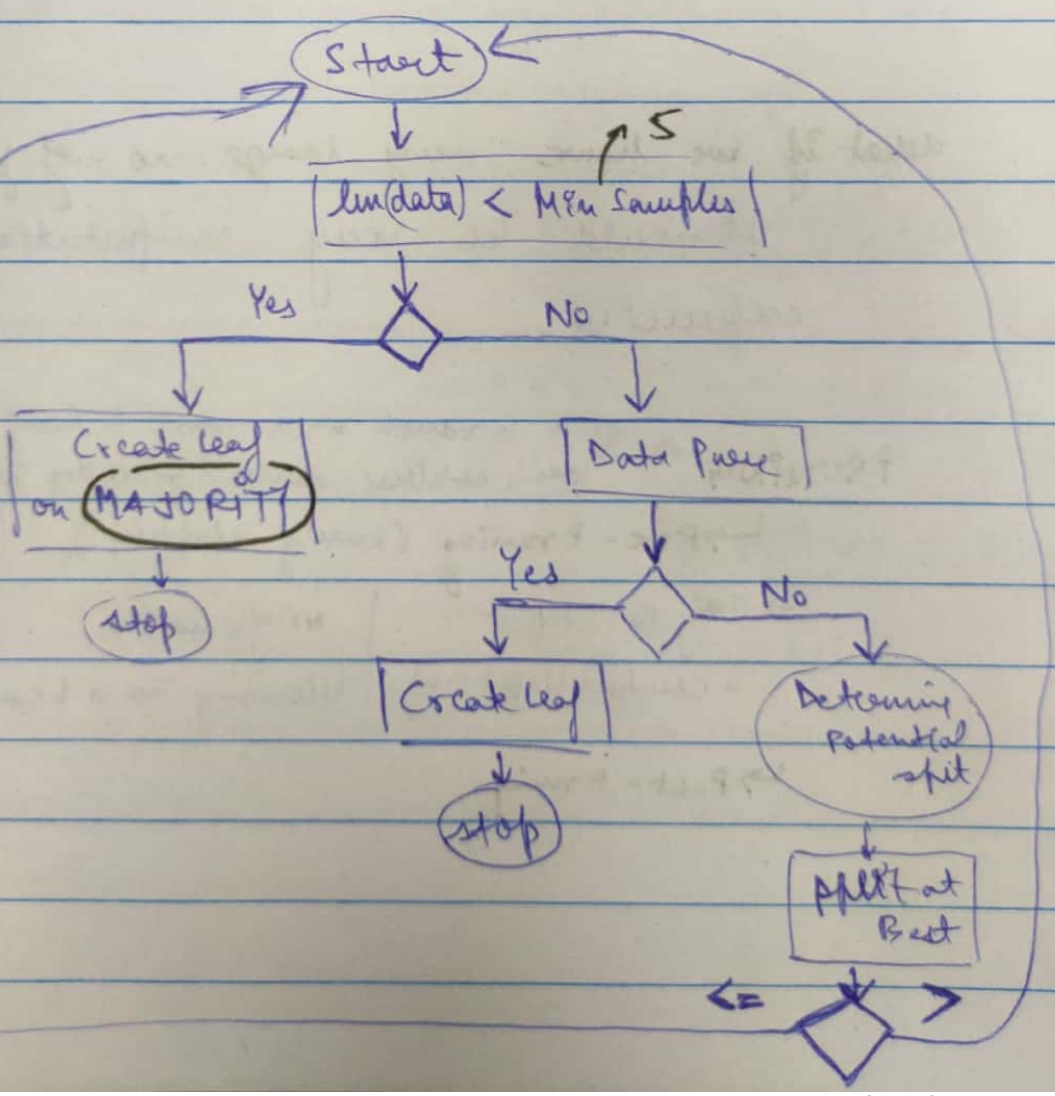
↳ Post-Pruning

Why can post-pruning help?

↳ Test set may not have some data points on which the training was done, that means subtrees for those are IRRELEVANT.

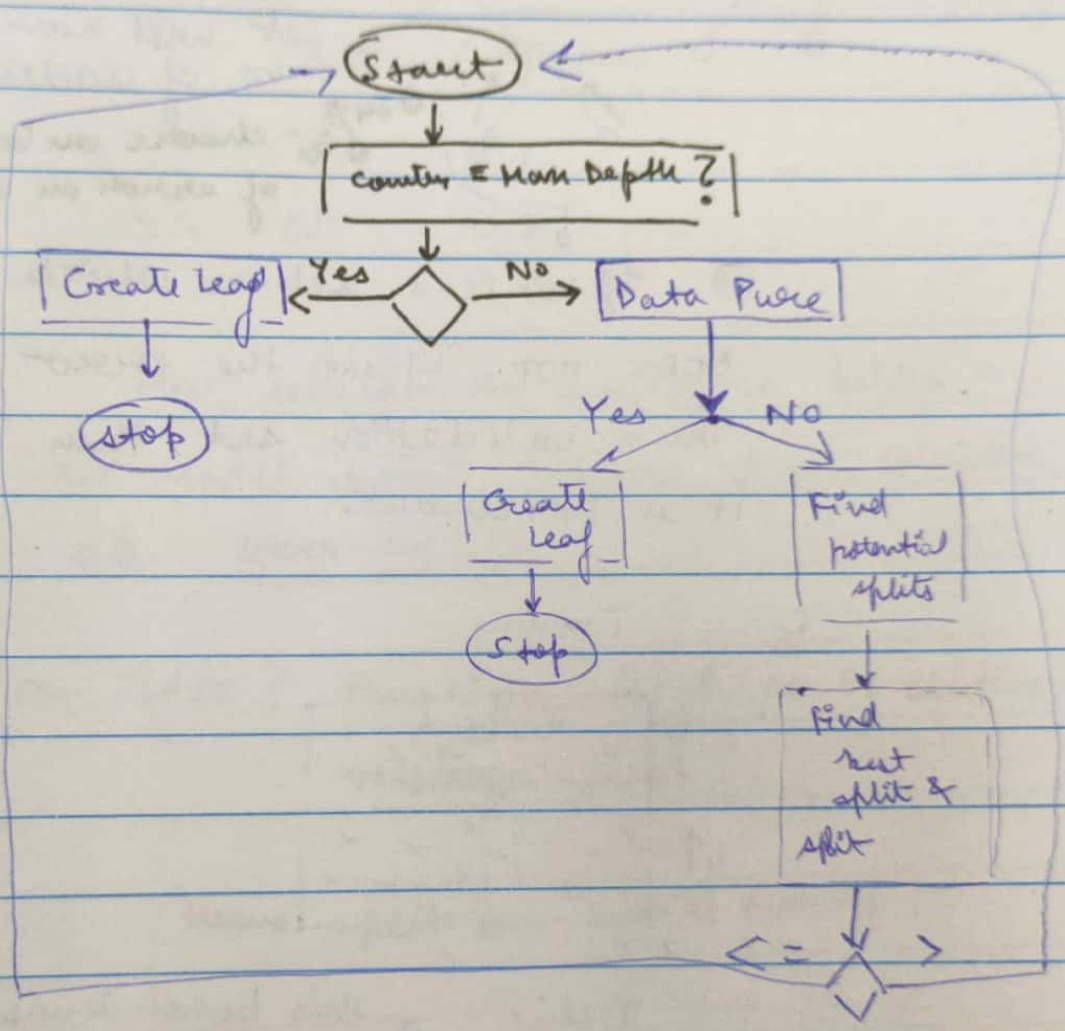
Some pre-pruning methodologies.

① Min samples. - To limit depth / decision boundaries.



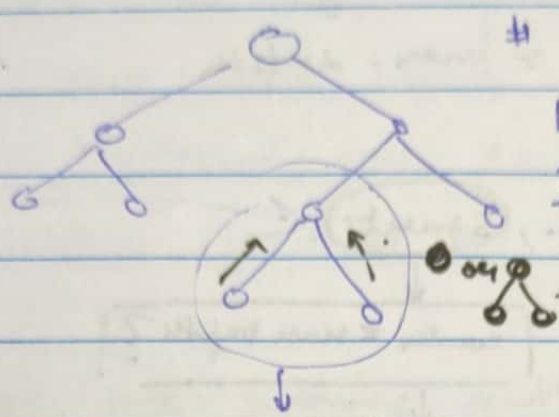
② MAX DEPTH

↳ Stop processing a branch if depth reaches > max. depth.



some post pruning methodologies.

4.



will start from Last LEVEL as it will have least no. of nodes.

- choose on basis of error on test set.

(#). If this is a subtree which DOES NOT affect the error on a validation set, then it is irrelevant.

Eg Training

Testing → this bunch is misclassified.

try removing this

SOME CAVEATS & DIFF. METHODS for SPLITS.

① What $IG_{A_i} = IG_{A_j}$?

If we have same Info Gains for eg.

	3 values	2 values
	A_1	A_2
IG	0.7	0.7

Here, selecting A_2 would be better as the model would become less complex at lower splits.

Better Idea ? Penalize High no. of distinct values in attrib.

Means split on GAIN RATIO

$$\text{Gain Ratio}_{A_i} = \frac{\text{Info Gain}_{A_i}}{\text{Entropy}_{A_i}}$$

② GINI INDEX – Calculate for any attribute t,

NOTE : A minimum ~~split~~ gini index is a preferable way for split.

Algos like CART, SLIQ, SPRINT..

Instance based learning.

Lazy learning

KNN Algorithm (K-Nearest Neighbours).

↳ Assigning the target features value/ class on the basis of the K-nearest points in the data set.

* Supervised learning.

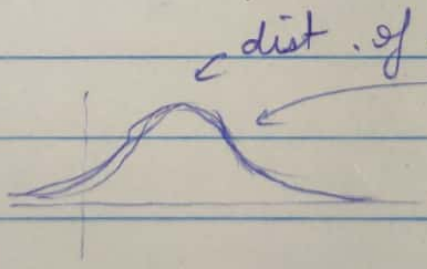
↳ Basis is a similarity measure (distance)

↳ Stores all available cases & classifies new cases based on a similarity measure.

There are 2 types of models -

① Parametric - When there is an underlying assumption for data to come from some distribution.

② Non-parametric - No assumptions.



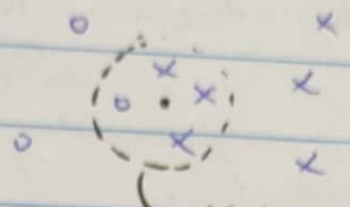
Eg

* M_1 has been told that the data has a gaussian distribution

M_2 → no assumption

⇒ KNN is a NON-PARAMETRIC model.

Eg We have taken $K = 4$,

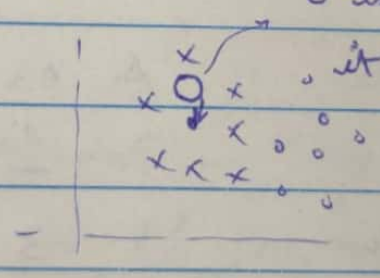


If K = 4, x is assigned to •.

How to choose K?

1. K should not be very small.

Why? To avoid classification due to noise.



O is assigned. although it was a noisy point.

K should be selected as < sqrt(n) something.

some distance measures →

Euclidean, Manhattan & Minkowski.

3-KNN : Ex 1

use the 3 closest profile for classification.

Customer	Age	Income	No. of credit cards	class.
		X		Y

↳ One thing to note is that NORMALIZATION of variables is important. why?

SCALE affects distances.

Eg.	A_1	A_2	A_1	A_2	
	2	1000	2	1	scaled down
	3	3000	3	3	
	1	4000	1	4	
	5	6000	5	6	

7 x

One way to standardize →

$$X_s = \frac{X_i - \min}{\max - \min}$$ → Min-Max scaler.

Another →

$$X_s = \frac{X_i - \text{mean}}{\sigma}$$ → Std. scaler.

CONS - computational time

↓

$O(N \log K)$ (for euclidean distance.) for finding the first k minimums out of the training set.

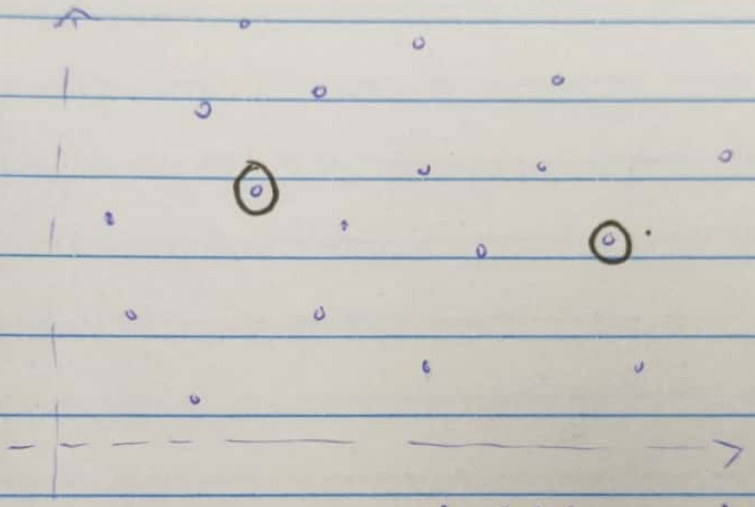
WORK AROUND — Try to reduce the no. of features of the dataset to reduce computational time.

Also, note that distance measures do not work with categorical vals. Need some data transformation steps.

unsupervised learning.

K-Means Clustering

Selecting the first K-centeroids & iteratively improving upon the cluster centers.



Init - Initialize K-data points