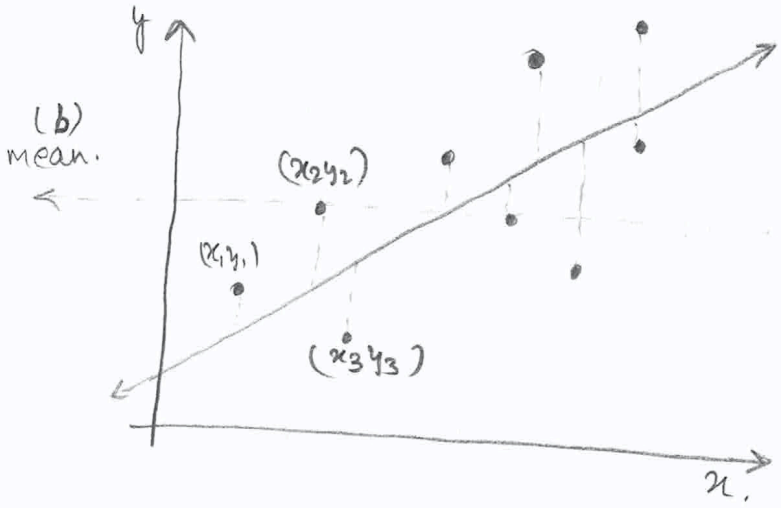
LINEAR REGRESSION / LEAST SQUARES

(b)
the 'mean' cuts through the average value of 'y' & gives us a good starting point to compare.



how well this line 'b' fits the data

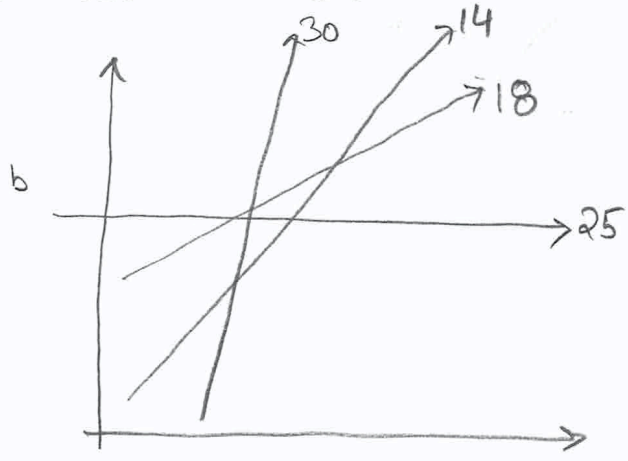
$= (b-y_1)^2 + (b-y_2)^2 + (b-y_3)^2 + \dots$

⇓

is call sum of Squared Residuals = 25

Residuals = difference b/w real data & the fitted line 'b'.

Rotating line 'b' gives better sum of sq. residual. upto a certain slope & gets worse later.



$y = b + ax$

b → y intercept

a → slope

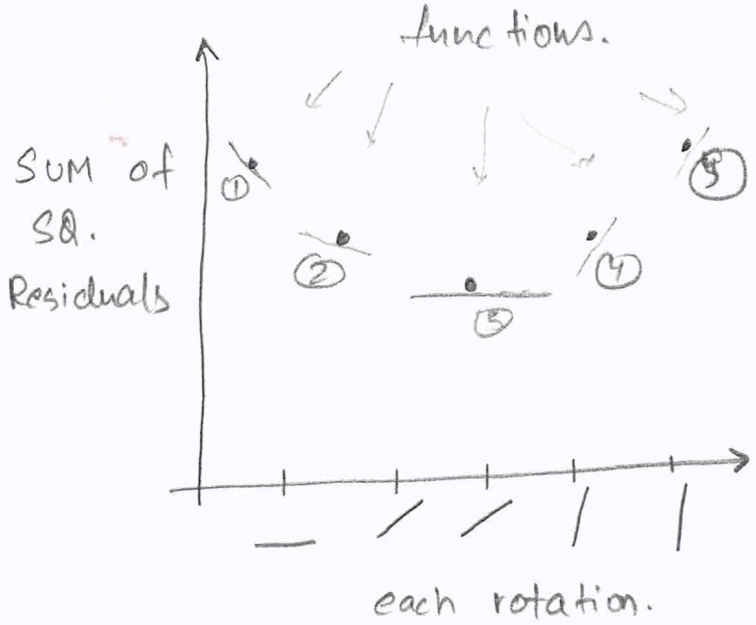
To Find the values of 'a' & 'b' so that we MINIMIZE the sum. of SQ. Residuals.

$= ((ax_1 + b) - y_1)^2 + ((ax_2 + b) - y_2)^2 + \dots$

$(ax_1 + b)$ → value of line at x_1

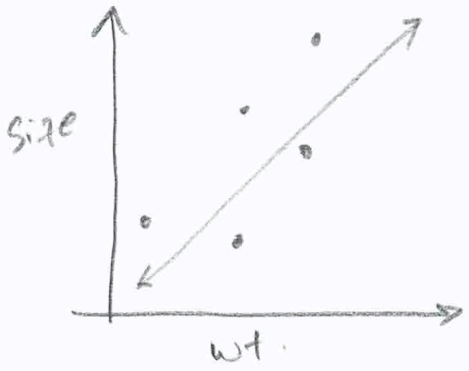
y_2 → observed value of x_2.

To find the optimal rotation of the line ⇒ take derivative of the functions '•' in graph



Derivatives give us the slope of the function at every point.

Function ① slope is steepest. & Function ③ has slope = 0 (least SQ.)

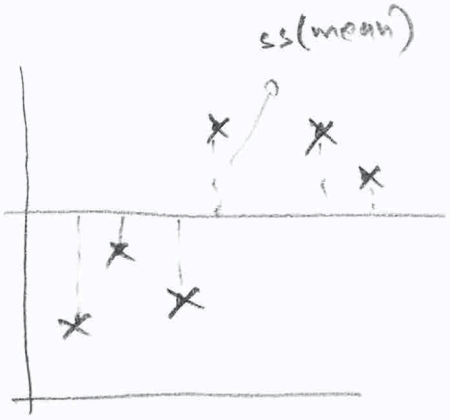


with this line

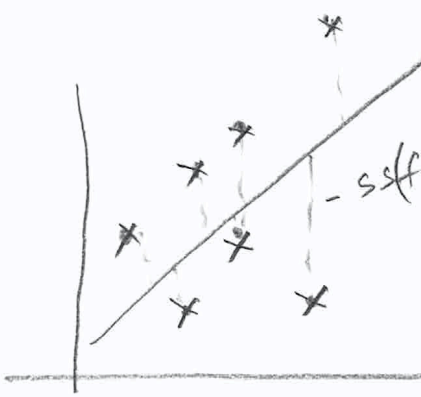
1) Calc. R^2 to find relation b/w size & wt. large value imply large effect.

2) Cal. p-value if R^2 is statistically significant

3) find size given wt.

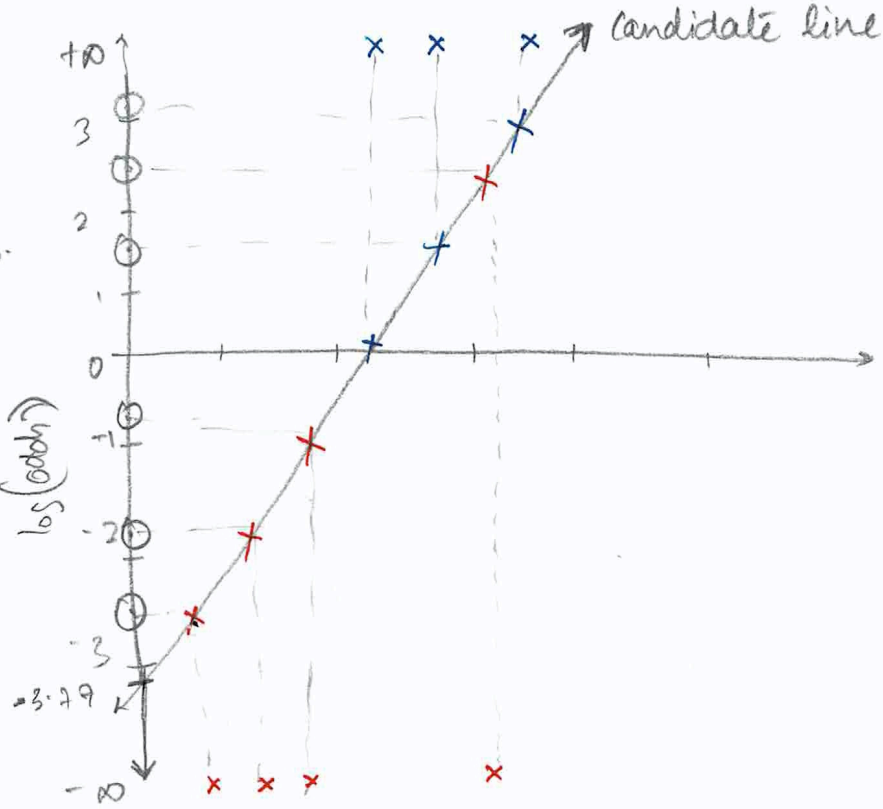


$$R^2 = \frac{SS(mean) - SS(fit)}{SS(mean)}$$



LOGISTIC REGRESSION: MAXIMUM LIKELIHOOD, R^2, p-value

Project the original data on candidate line. These points are log (odds) pts.

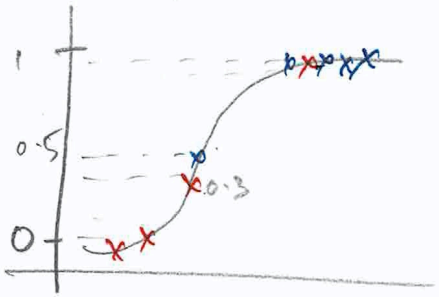


Transform log (odds) to

$$\Downarrow$$

$$p = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$

$$\Downarrow$$



× - likelihood that these are obese ⇒
0.45, 0.9, 0.91, 0.92 × ...

× - likelihood that these are NOT obese =
(1 - pr(being obese)

= (1 - 0.9), (1 - 0.3), (1 - 0.01), ...

likelihood of data points on squiggle line $= (0.49 \times 0.9 \times 0.91 \times 0.92) \times [(1-0.9) \times (1-0.3) \times (1-0.01)]$

we take log & find MAX value for this

$$\text{Max } \log\left(\text{likelihood of data on squiggle line}\right) = \log(0.49) + \log(0.9) + \log(0.92) + \log(1-0.9) + \log(1-0.3) + \ldots$$

$$= -3.77$$

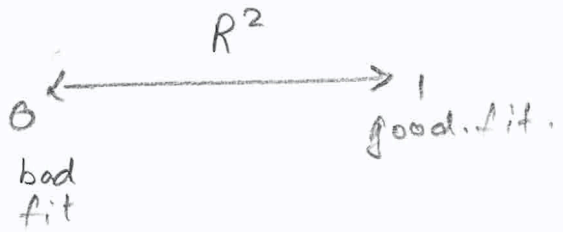
$LL(fit)$ = log (likelihood of data on squiggle line) = -3.77

probability of obesity = $\frac{\text{no. of obese}}{\text{total}} = \frac{\times\times\times\times}{\times\times\times\times\times\times\times\times\times} = \frac{5}{9} = 0.55$

= LL (probability).

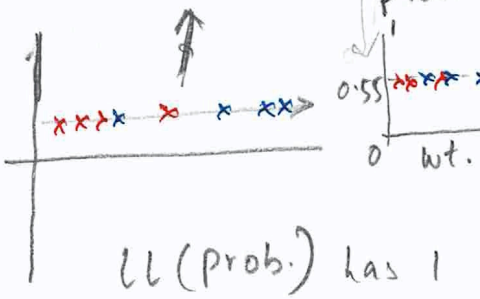
$$R^2 = \frac{LL(\text{prob.}) - LL(fit)}{LL(\text{prob.})}$$ ← R^2 from log likelihoods.

LL(prob.) = LL(overall prob.)



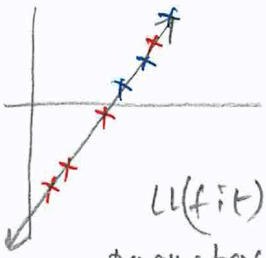
A Chi-Square value $= 2\left[LL(fit) - LL(\text{prob.})\right]$

log (obese / not obese) = translate back to prob.



LL (prob.) has 1 parameter → y-intercept

Deg of freedom:

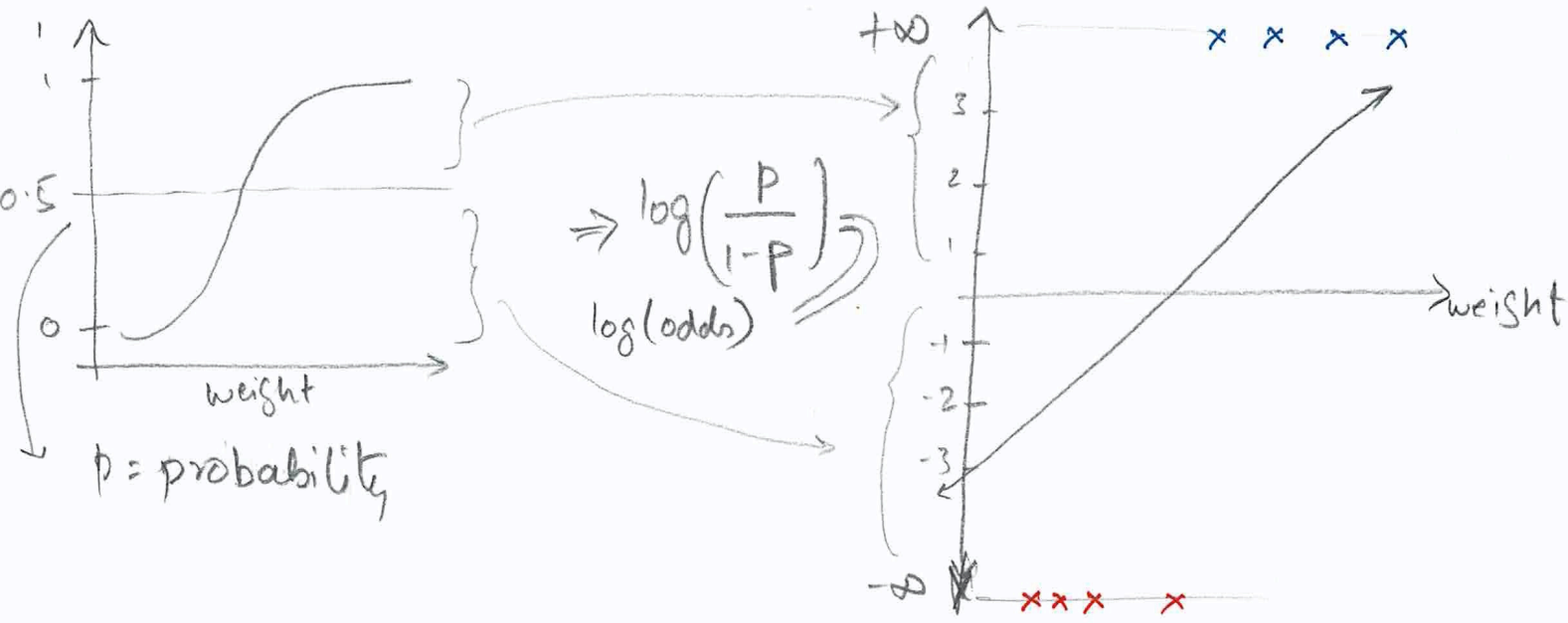


LL(fit) has 2 parameters - slope, intercept

$\therefore df = 2 - 1$

p-value = (Chi sq. val. & df)

LOGISTIC REGRESSION - COEFFICIENTS.



The coefficients are represented in terms of log (odds) graph

$$y = -3.48 + 1.83 \times (\text{weight})$$

↓ intercept ↓ slope

Python Result:

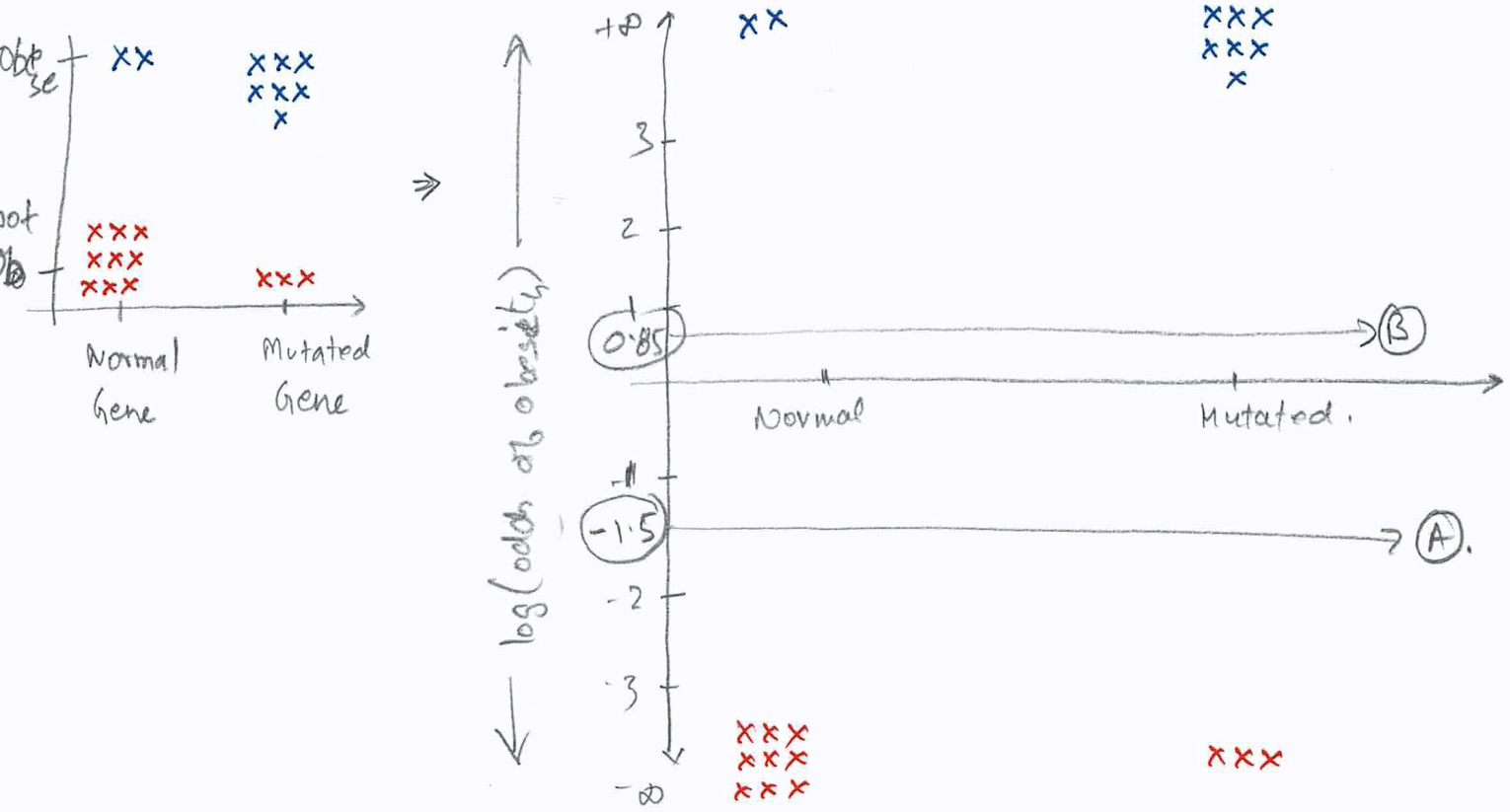
Coefficients:

	Estimate.	Std.Error	z-value	Pr(>|z|)
(intercept)	-3.48	2.36	-1.47	0.14
weight	1.83	1.08	1.67	0.09

z-value = $\frac{\text{estimate}}{\text{std. error}}$ $\Rightarrow$ (no. of Std. Dev. the estimated intercept is away from ϕ) $\Rightarrow$ wald's test.

Since its less than 2 std.Dev. away from ϕ, its not statistically significant. & this is confirmed by p-value. (large)

For every unit of wtg. gained, log(odd of Y/N) increases by 1.83.



$$\log(\text{odds})_{\text{Normal}} = \log\left(\frac{2}{9}\right) = -1.5 \rightarrow (A)$$

$$\log(\text{odds})_{\text{mutated}} = \log\left(\frac{7}{3}\right) = 0.85 \rightarrow (B)$$

} t-test

$$\text{Size} = (A) \times B_1 + ((B) - (A)) B_2 = -1.5 B_1 + 2.35 B_2$$

Coeff:	Estimate	Std. Error	z-val.	Pr(>|z|)
(intercept)	-1.5	0.78.	-1.92	0.054
Mutated	2.35	1.04	2.25	0.024.

log (odd-ration) in log scale that tells how much having the 'mutated' ↑/↓ the odds of being obese.

greater than 2 ∴ statistically significant.

LOGISTIC REGRESSION.

Logistic Reg. predicts if something is T/F, Y/N.

- This curve tells the probability of a 0/1 given wt. Although this helps us the probility, its usually used for clanification.

Eg. when prob > 50% => '1' else '0'.

linear Regression

least SQ's. & R^2

Logistic Regression

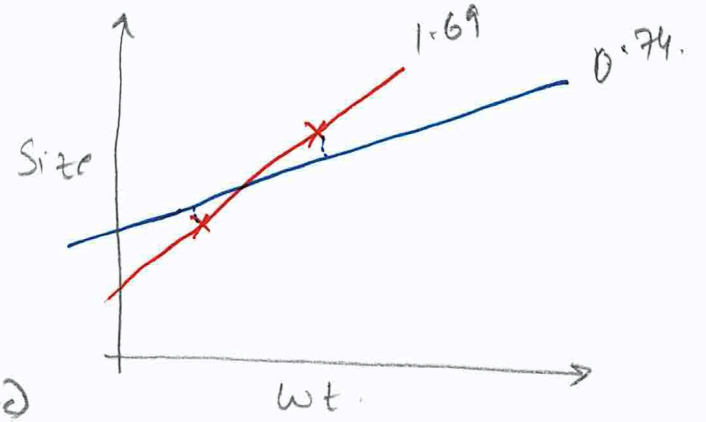
Prob. of Y/N.

Cal. Curve with Maximum likelihood.

Generalited linear Models (GLM).

LASSO & RIDGE REGRESSION; ELASTIC RIDGE

By introducing small amount of Bias, we can significantly drop Variance.



least sq. = y-intercept + slope × wt.

(linear) (logistic)

Ridge = Sum of sq. residuals/likelihood + $\lambda \times \text{slope}^2$

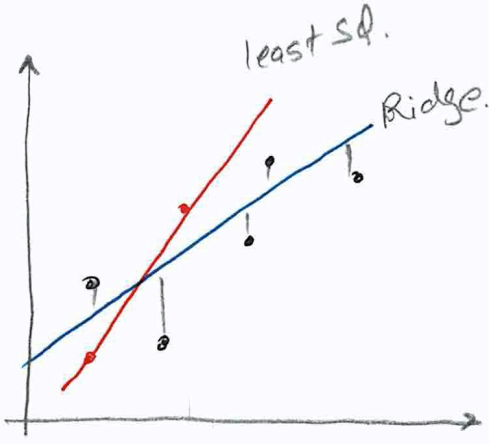
Ridge regression contains all parameters except, y-intercept.

How to determine the best 'λ' value? ?

Size = y-intercept + slope × wt + dist difference × highfat + speed × Airspeed ...
+ λ (slope^2 + dist diff^2 + speed^2 + ...)

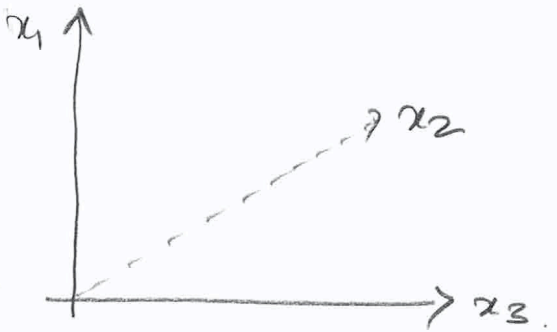
Advantage:

When the sample size are relatively small, the Ridge Regression can improve prediction (made from new data (reduce variance) by making predictions less sensitive to train data by adding Penalty to the thing that must be minimized. →



Even when there isn't enough data to find least Sq. parameter estimate, the Ridge Regression can still find Solution with Cross Validation & penalty.

For least Sq. - since there are 3 parameters, ideally we should have 3 data pts. But this is not a requirement for Ridge.



LASSO = Sum of sq. residuals + $\lambda \times |\text{slope}|$

The big difference b/w Lasso & Ridge ⇒ Ridge can only asymtotically shrink the slope to zero but in lasso, slope = 0.

Eg: $\text{Size} = C + m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + \cdots$

when x_3 & x_4 are not significat, Ridge cannot make m_3 & $m_4 = 0$ but lasso can exclude useless variables from Eq.

Lasso works best when there are lot of useless variables

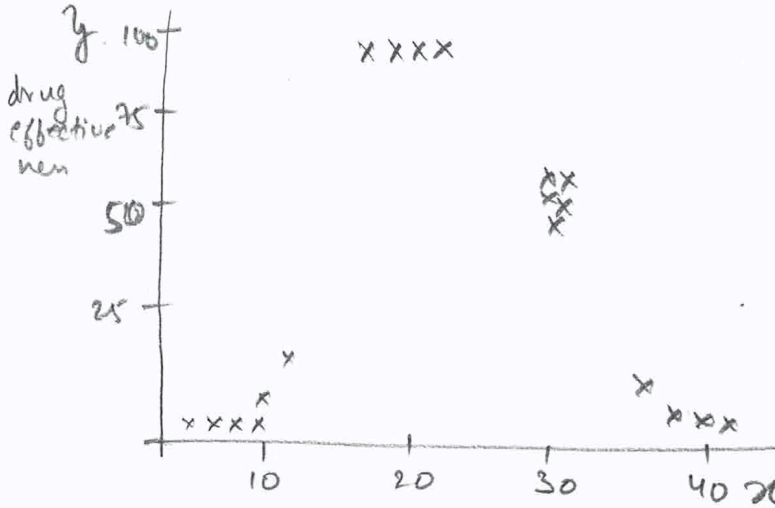
Ridge works best when most of the variables are useful

Elastic Net = Sum of Sq. residuals + $\lambda_1 \times |\text{slope}| + \lambda_2 \times \text{slope}^2$ (lasso, ridge)

Elastic Net - groups & shrinks the parameters associated with correlated variables & leaves them Eq. or Removes them all at one Reomd. when the no of parameters are in Millions

REGRESSION TREE

1) Its difficult to apply linear Regression line on this data set. ∴ regression tree.



2) We determine how to divide the 'x' by trying different thresholds of x & calculating the sum of sq. residuals. (SSR) & pick the one with MIN. SSR. This becomes the Root Node

3) When there multiple x, we pick x_i with least (SSR) as Root Node.

PRUNE: prone to over fit the regression tree.

1) Calculate the tree score

$$\text{Tree score} = SSR + \alpha T$$

T → no. of terminal nodes (leaf)

α → tree complexity penalty

lowest SSR using Cross validation

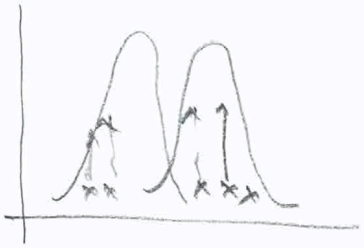
SATURATED MODELS & DEVIANCE - LOGISTIC REGRESSION

the normal curve is a model of the data.
We use this instead of the original data to Calc. the estimated probabilities & statistical tests.

Likelihood of data Null Model = $0.2 \times 0.5 \times 0.8 \times 1.5 \times \ldots$

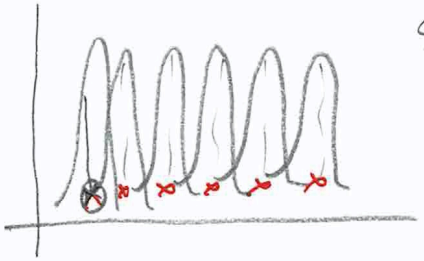
$= 0.03$.

LL (Null Model) = $\log(0.03) = -3.51$



No of parameters = 2

LL (Proposed Model) = 3.75



Saturated Model = makes out no. of parameters we can use. ie, one parameter per data pt.

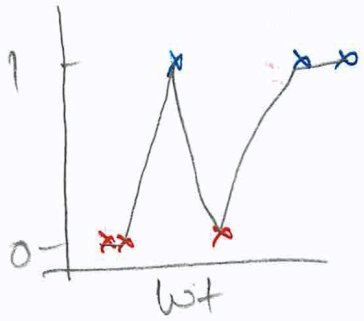
LL (Saturated Model) = 1291.5

Residual Deviance = $2\left[LL(\text{Saturated model}) - LL(\text{Proposed})\right]$

↳ 2 times makes these LL difference to have Chi-sq. distribution. with df = no of parameters.

Null deviance = $2\left[LL(\text{saturated}) - LL(\text{Null})\right]$

R^2 = Null deviance - Residual Deviance.



In logistic Regression, the Saturated Model perfects fits the squiggle.

$$\therefore R^2 = \frac{LL(\text{null}) - LL(\text{Proposed})}{LL(\text{Null})}$$

&

$$\text{Residual deviance} = 2 \times -LL(\text{Proposed})$$

$$\text{Null deviance} = 2 \times -LL(\text{Null})$$

K-NEAREST NEIGHBORS.

A simple way to clarify data.

When $K = 5$, we assign the point to the cluster category that gets "MAX. votes"

Red - 6, Blue - 2

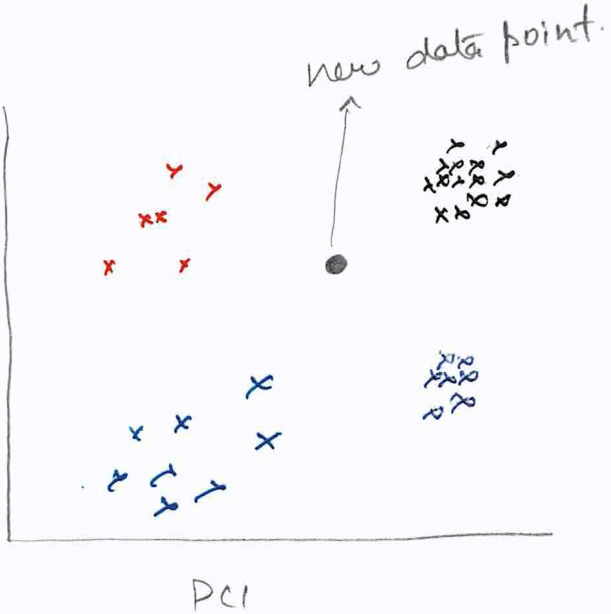
Black - 14, Blue II - 7.

Since Black got most votes, its assigned to Black cluster category.

When the point is b/w 2 categories -
if we can avoid. when K = odd number.

K-NEAREST NEIGHBORS.

A simple way to clarify data.



When $K = 5$, we assign the point to the cluster category that gets "MAX. votes"

Red - 6, Blue - 2

Black - 14, Blue - 7.

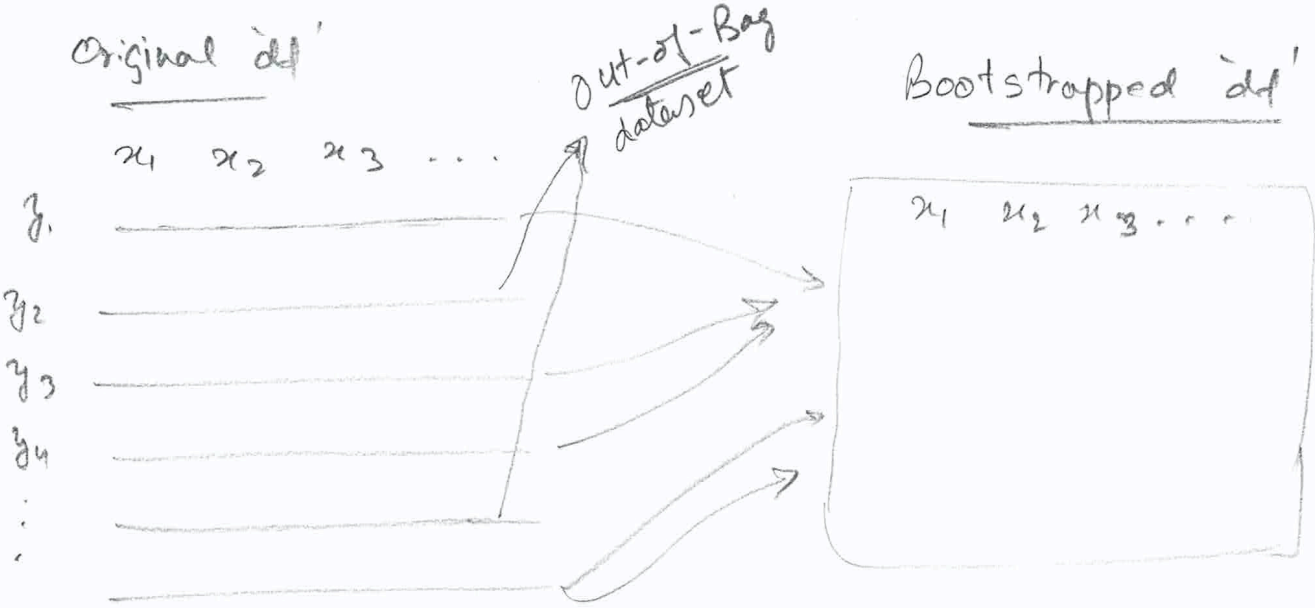
Since Black got most votes, its assigned to Black cluster category.

When the point is b/w 2 categories -
if we can avoid. when K = odd number.

RANDOM FORESTS

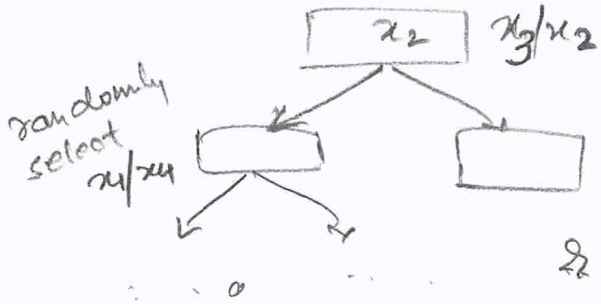
① From the original 'df', randomly select samples ⇒ Bootstrapped dataset.

* A sample can be picked more than once.



typically 1/3 of data do not end up in bootstrapped df.

② Create a decision tree on bootstrapped df but use only random variables ($x_1, x_2 \ldots$) i.e. select only 'n' variables instead of all x_i.



← x_1 x_2 x_3 x_4.

& so on... to build the tree. by considering only random variables at each node.

① Using a bootstrapped sample & considering only a subset of variables at each node/step gives wide variety of trees.

The Variety make Random forest more effective when compared to Decision trees.

We take this data & run it down all the trees in RF to & choose one with max. votes.

x_1	x_2	x_3	x_4	
y_{new}	N	98	?	Yes.

x_4	
Y	N
5	2

[Bootstrapping the data AND using aggregate to make decision is called BAGGING.]

The out-of-Bag was not used to create the tree's ∴ this df is run across all trees that were built.

Accuracy of RF - proportion of out-of-Bag samples that were correctly classified.

The samples that were incorrectly classified ⇒ Out-of-Bag Error

How to choose No. of x initially? -

Compare Out-of-Bag Error for a RF built using 2 variables per step (vs) 3 variables per step.... & choose the most accurate RF.

Summary:

change No. of variables
1) Build a RF (typically start with $\sqrt{x}$ of no. of variables)
2) Estimate Accuracy of RF

DECISION TREE

(A) Root nodes; Internal nodes; Leaf nodes;

x_1 Chest pain	x_2 BP	x_3 Blocked arteries	y Heart Disease (Y)
N	N	N	N
Y	Y	Y	N
Y	N	Y	N
.	.	.	.

1) How to determine the Root Node x_1, x_2, x_3.

2) Take each x & calculate the "Gini Impurity" & pick the 'x' with least Gini impurity valu

Gini = 0.36 (for x_1): Y N → 100 39; Y N → 34 125

Gini = 0.30 (for x_2): Y N → 37 127; Y N → 100 33

Gini = 0.38 (for x_3): Y N → 9 10; Y N → 100 125

$\Rightarrow$ x_2 $\Rightarrow$ Root Node.

3) To pick Inter node - repeat the same procen & pick the 'x' with least impurity.

4) If the Node itself has lowest impurity, it becomes leaf node

(B) Wt.(x_1) & $\Rightarrow$

5
2
4
7
.

1) Sort x_1 - low → high.
2) Cal. the avg. b/w x_i
3) Cal. the Gini impurity for each Avg. x_i

(C) for Ranked x_i like, 1-4, multiple choice → Cal. the Gini impurity at each value of x_i & pick the lowest

Decision trees have downside of often being overfit.

Missing data - a) find the co-relation b/w the missing column & another column in df. Use liner Regression to predict the value of missing value; or choose mean/median

b) for categorical - choose the category with max. value; ie in (Y=8, N=10), substitute with 'N'; or find the column with highest co-relation & predict the value of missing data