DATE: / /
PAGE:

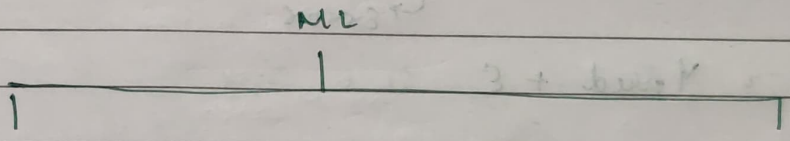
Supervised Learning - Regression (SLR)

$y = mx + c$ or $c + mx$

$y = b_0 + b_1 x$

b_0 → Constant (Intercept of best fit line with y-axis)

ML - Machine identifying pattern without explicitly coding.



Supervised learning

↓

There is a Target variable to be predicted

↓

Regression

Target variable is (continious) numeric

Classification

Target variable is categorical.

Unsupervised learning

↓

No target variable to be predicted.

↓

Understand if there is a group in data

↓

Clustering.

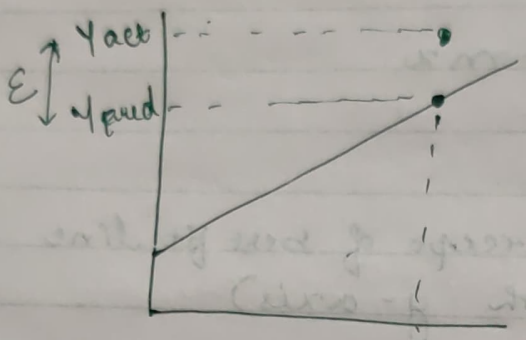
Target variable - dependent variable, - response variable or y - all are same.

Independent - Explanatory variables - X_s - all are same
Predictor variable

$\hat{y} = b_0 + b_1 x_1$

$\hat{y}$ → y-hat → predicted y → y-pred

y actual b_0 & b_1 are parameters of Linear Regression



Error : $y_{actual} - \hat{y}$ (pred)

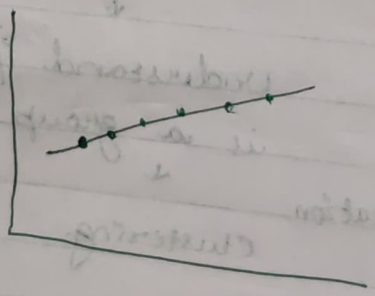
$y_{act} = b_0 + b_1 x_1 + \varepsilon$

↳ Error

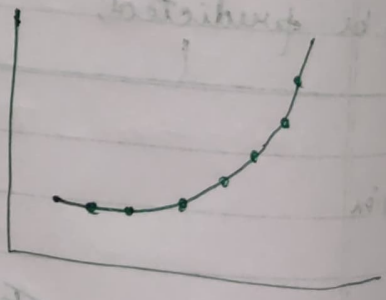
$y_{act} = y_{pred} + \varepsilon$

Types of Association

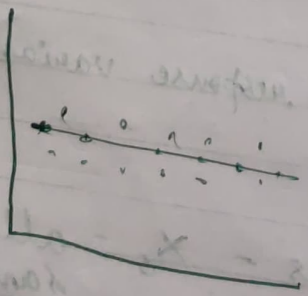
1) Linear



2) Non-Linear



3) No relationship



DATE: / /
PAGE:

How to draw best fit line :-

1) Ordinary Least Squares Method.

Best fit line - line which gives us lowest sum of squares of Error.

$$\text{Min of } \Sigma \text{ error}^2$$

↳ Cost function or Loss function.

Solving partial derivatives

$$\frac{\partial \text{ Cost func}}{\partial b_0} = 0 \quad - (1).$$

$$\frac{\partial \text{ Cost func}}{\partial b_1} = 0 \quad - (2)$$

gives

$$b_1 = \frac{\text{Cov}(x, y)}{\text{var}(x)}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

Business interpretation of b_1

For every unit increase in x_1, $\hat{y}$ will increase by b_1.

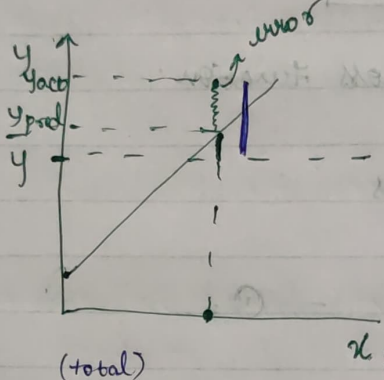
b_0 is not always interpreted. Depends on domain knowledge.

We use partial derivatives, as derivatives dont work on more than 2 unknowns.

Performance Measure

How close the predicted value is compared with Actual.

- Higher the value of R^2, better the prediction. Its always b/w 0 & 1
- proportion of variation of y explained by the model is coefficient of Determination - R^2



— → Actual variation in the data (dist b/w mean & data)

— → variation that model thinks is happening (dist b/w mean & pred data)

~ → Error.

$$SST = \sum (y_{act} - \bar{y})^2 \quad \text{(total)}$$

R - coeff of correlation

$$SSR = \sum (y_{pred} - \bar{y})^2 \quad \text{(regression)}$$

$$SSE = \sum (y_{act} - y_{pred})^2 \quad \text{(error)}$$

$$\boxed{R^2 = \frac{SSR}{SST}}$$

$$SST = SSR + SSE$$

$$SSR = SST - SSE$$

$$\frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$\boxed{R^2 = 1 - \frac{SSE}{SST}}$$

Sometime

Error is also known as Residual (SSR)
- Regression (green) is known as explained (SSE)
  ↳ variation explained by the model

DATE: / /
PAGE:

** Problem with R^2 - R^2 goes up even if there is a useless column
In that case, we do adjusted R^2.

$$\text{Adj } R^2 = 1 - \frac{(1-R^2)(n-1)}{(n-k-1)}$$

n = no of obs
k = no of X_s (columns)

- Higher the better.
- If k goes up, Adj R^2 comes down

Adj R^2 is derived from $= 1 - \frac{SSE/n-k-1}{SST/n-1}$

→ Used Adj R^2 :- when you are supposed to add/remove columns. (or to decides if its useful or not).

df of SSR (or model) = k
df of SST = n-1
df of SSE = n-k-1.

.fit() - Trains the Algorithm.

- Reject zone for f-statistic (ANOVA) is always on right side.
- Larger the f-statistics, we may end up in rejecting Null hypothesis.

$$\text{f-stat} = \frac{SSR/k}{SSE/(n-k-1)} \rightarrow \text{df of SSR}$$

To find the p value from f-stats
scipy.stats → stats.
stats.f.sf(f_stat, df of num = k, df of den = n-k-1)

Testing if regression is valid/not :- F test.

H_0 : All coefficients = 0 → implies regression is not valid

H_a : All coefficients $\neq 0$ → regression is valid
At least 1

To check if a column is useful/not :- T test.

H_0 : coefficient of that $X_s = 0$ - implies that column is not significant.

H_a : coefficient of that $X_s \neq 0$ - column is significant

t stat = $\frac{\text{coeff}}{\text{std error}}$

→ Look at pval for columns (not for const) and determine if H_0 is rejected/Accepted.

→ If only 1 x, its simple linear Regression.
→ If more x, its multi linear regression

Modelname.resid → gives errors

Day 2

Assumptions of linear regression

- Tests before model building
1) Dependent variable must be numeric
2) Predictors must not show multicollinearity

Multicollinearity :- Correlation b/w X_s.
A strong correlation b/w X_s is not good for prediction

DATE: / /
PAGE:

Multicollinearity detection.

1) Determinant of the correlation Matrix } checks presence of Multicollinearity
2) Condition Number

1) Let D be determinant of corr Matrix
   $0 < D < 1$
   $D = 0$ high multicollinearity
   $D = 1$ No "

2) $CN > 1000$ Severe multicollinearity
   $100 < CN < 1000$ Moderate "
   $CN < 100$ No "

3) Correlation Matrix } which variables are involved in multicollinearity
4) Variance Inflation ~~Number~~ Factor (VIF)

VIF

Runs regression internally to check which variables have multicollinearity.

Target	Pred	R^2	$VIF = 1/(1-R^2)$
X_1	$X_2 X_3 X_4$		
X_2	$X_1 X_3 X_4$		
X_3	$X_1 X_2 X_4$		
X_4	$X_1 X_2 X_3$		

$$VIF = \frac{1}{1-R^2}$$

If $VIF >= 5$, there is strong multicollinearity that affects the model

Autocorrelation :-

Correlation with itself.
Is my record dependent on previous record

Durbin - Watson Test : (To check if it autocorrelated / not)

H_0 : The error terms are not autocorrelated
H_a : The error terms are autocorrelated.

If Test statistics = 2 - No autocorrelation
Practically exact 2 doesn't come. So take a range of 1.5 - 2.5

$0 <$ Statistics < 2 Positive autocorrelation

$2 <$ Statistics < 4 Negative autocorrelation

Homoscedasticity Assumption (errors should be homo)

Homo scedasticity
↓ ↓
Same Variance

Heteroscedasticity — Funnel type shape when plotted

We use Breusch Pagan Test

H_0 : The error terms are homoscedastic.
H_a : The error terms are heteroscedastic.

Jarque - Bera Test for Normality

H_0 : Data is Normal
H_a : Data is NOT Normal

Test performance is imp.

DATE: / /
PAGE:

** Even if VIF > 5, do not drop categorical variables

Interaction Effect :-

→ particular variable's effect on combination of variables.

Eg:- Effect of experience alone on Salary
" " Education " "
Effect of both experience & Education on Salary is even more high.

→ If intercept is needed, then only use sm. add_constant.
Else build the model without that.

$$\text{Mean abs percent error} = \frac{1}{n} \sum \left| \frac{Act - Pred}{Act} \right|$$

Overfitted :
Significance difference b/w RMS of Test & Train implies overfitted. → [Train is high, Test is low]

Note :- Scaling does not affect/impact Regression

Feature Selection :-

1) Forward selection
- Start with Null model and then go on adding columns one by one.

2) Backward selection
- Start with full model & then go on dropping columns

Stepwise Regression:-

→ Combination of both fwd selection & backward elim

Automated :-

SFS - Sequential feature selector (wrapper method)

RFE - Recursive feature eliminator

Validation

K fold cross validation.

Total 1000 rows.

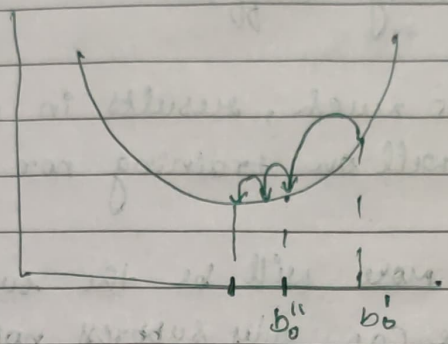
$X_1 \dots X_n$	y
200	A
200	B
200	C
200	D
200	E

Train. R^2	Validate (test). R^2
ABCD	E =
ABCE	D =
ABDE	C =
ACDE	B =
BCDE	A =
	Avg =

Cross validation Score

$y = \underset{M}{M}x + c$

Leave one out cross Validation (LOOCV)
Gradient descent



Regularization:
(Eg of teaching a child about dog (breeds)).
We know that,
LR :- Minimize $\sum (Y_{Act} - b_0 - b_1 x_2 \cdots)^2$,
↳ results / possibility of overfitting.

1) So,
Minimize $\sum (Y_{Act} - b_0 - b_1 x_2 \cdots)^2 + \lambda \sum_{i=1}^{n} \beta_i^2$

In python, $\lambda = \alpha$, β = coefficients = b_0, b_1

Algorithm → Ridge Regression (L2 regularization)
↳ Used when there is non-linear relationship between the X_s.

2) Lasso Regression (L1 regularization)

Minimize $\sum (Y_{act} - b_0 - b_1 x_2 \cdots)^2 + \lambda \sum_{i=1}^{n} |\beta_i|$
β → coeff = $b_0, b_1, b_2 \cdots$

$\lambda \rightarrow$ Hyper Parameter

↳ We decide this Eg it is the parameter we fo
through Algorithm.

→ As λ increases, penalty increases.
effect - lines with small coeff gets selected.
(suppressing coeff).

→ Increasing λ too much, results in underfitting
Neither works well on training nor Testing.

→ Larger the α/λ, more will be the suppression
in Ridge. (Ridge can only suppress not drop)

→ Even if there is multicollinearity in columns, Ridge
and Lasso can be applied.
→ If there is insignificant columns, Lasso drops it
automatically.

Elastic Net regression :-

Minimize $\sum (y_{Act} - b_0 - b_i x - \dots) + \lambda_{ridg} \sum b_i^2 +$

Theoretically → $\lambda_{Lasso} \sum |b_i|$

Practically → $\lambda = \lambda_{Ridge} + \lambda_{Lasso}$

λ ratio = $\dfrac{\lambda_{Lasso}}{\lambda_{Ridge} + \lambda_{lasso}}$

Algorithm used inside sklearn linear Regression
is Gradient Descent

DATE: / /
PAGE:

Grid Search (Used to find) tune Hyperparameter

1) If there are large no of columns and you feel many are useless & ~~inter~~, go for lasso and also interrelations

Supervised Learning:

Classification

- Dependent / Target is categorical variable.
- If means are 1.96 SD away from each other, then it is considered to be significant
- For Non linear model, there are no assumptions.

Gradient Descent :- (Iterative Approach)

$$\Delta \beta \propto - \Delta \text{Error}$$

(or) $$\Delta \beta \propto \frac{1}{\Delta \text{Error}}$$

$$\Delta \beta = - \eta \Delta \text{Error}$$

η - Learning Rate

$$E = \frac{1}{2}(y - \hat{y})^2$$

$$\frac{1}{2}(y - (\beta_0 + \beta_1 x))^2$$

differentiate : $\frac{1}{2} \cdot 2 (y - (\beta_0 + \beta_1 x))$

diff wrt β_0 $-(y - \hat{y})$

$$\Delta \beta_0 = \frac{\partial E}{\partial \beta_0} = -(y - \hat{y})$$

$$\Delta \beta_1 = \frac{\partial E}{\partial \beta_1}$$

$$= \frac{1}{2} \cdot 2 \left(y - (\beta_0 + \beta_1 x) \right)$$

diff wrt β_1

$$\Delta \beta_1 = -(y - \hat{y}) x$$

As error gradient is smaller, it will approach the best fit line

- Once the SSE reaches saturation - 2 consecutive iteration's SSE difference is insignificant (stopping criteria)
- Once the SSE starts increasing, we get to know that, we are going away from best fit line (values) there we stop.

Drawback

- Tends to underfit - full batch gradient - high bias error
- Tends to overfit - SGD - high variance error

Mini batch learning → Non linear model.

Logistic Regression:- (Sigmoid)

$\hat{y}$ of classification should give probability score

$$\hat{y}_{Prob} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Also called as Soft Limit function

DATE: / /
PAGE:

Error Gradient $= \frac{1}{2}\left(y - \frac{1}{1+e^{-(\beta_0+\beta_1 x_1)}}\right)^2$

$$\frac{\partial E}{\partial \beta_0} = \frac{1}{2} \cdot 2\left(y - \frac{1}{1+e^{-(\beta_0+\beta_1 x_1)}}\right)^1$$

$$= \left(0 - \left(\frac{(1+e^{-(\beta_0+\beta_1 x_1)})(0) - (1)(0+e^{-(\beta_0+\beta_1 x_1)}(-1))}{(1+e^{-(\beta_0+\beta_1 x)})^2}\right)\right)$$

$$= \frac{e^{-(\beta_0+\beta_1 x_1)}}{(1+e^{-(\beta_0+\beta_1 x)})^2}$$

$$\boxed{\frac{\partial E}{\partial \beta_0} = -\hat{y}(1-\hat{y})(y-\hat{y})}$$

$$\frac{\partial E}{\partial \beta_1} = -(y-\hat{y})\,\hat{y}(1-\hat{y})\,x$$

If probability score $> 0.5 \Rightarrow$ class $= 1$

$< 0.5 \Rightarrow$ class $= 0$

Concept of Odd's

Mutually Exclusive $\rightarrow$ Union [Addition.

$$\boxed{\text{Odd's} = \frac{P(\text{occurring})}{P(\text{non occurring})}}$$

odd's ratio is Relative Risk.

- biased coin $P(H) = 0.7$

$$O(H) = \frac{0.7}{0.3} = 2.33.$$

fair coin $P(H) = 0.5$

$$O(H) = \frac{0.5}{0.5} = 1$$

∴ odd's of head in biased coin is 2.33 times the odd's of fair coin.

SE - side effect

Coraxin

$P_{SE} = 0.2$

$$O(P_{SE}) = \frac{0.2}{0.8}$$

$$O(P_{SE}) = 0.25$$

Covisheild

$P_{SE} = 0.1$

$$O(P_{SE}) = \frac{0.1}{0.9}$$

$$O(P_{SE}) = 0.11$$

$$\frac{O_{cx}(P_{SE})}{O_{cs}(P_{SE})} = 2.3$$

odds ~~Prob~~ of side effects of Coraxin is 2.3 times the odds of side effects of Covisheild.

$$\hat{y}_{Prob} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \text{ ---> let this be } z.$$

$$= \frac{1}{1 + e^{-z}} \times \frac{e^z}{e^z} \quad (\text{to make } e^{-z} \text{ as } e^z)$$

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

DATE: / /
PAGE:

$$\frac{\hat{y}_{prob}}{1-\hat{y}_{prob}} = \frac{\frac{e^z}{1+e^z}}{1-\hat{y}_{prob}}$$

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

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

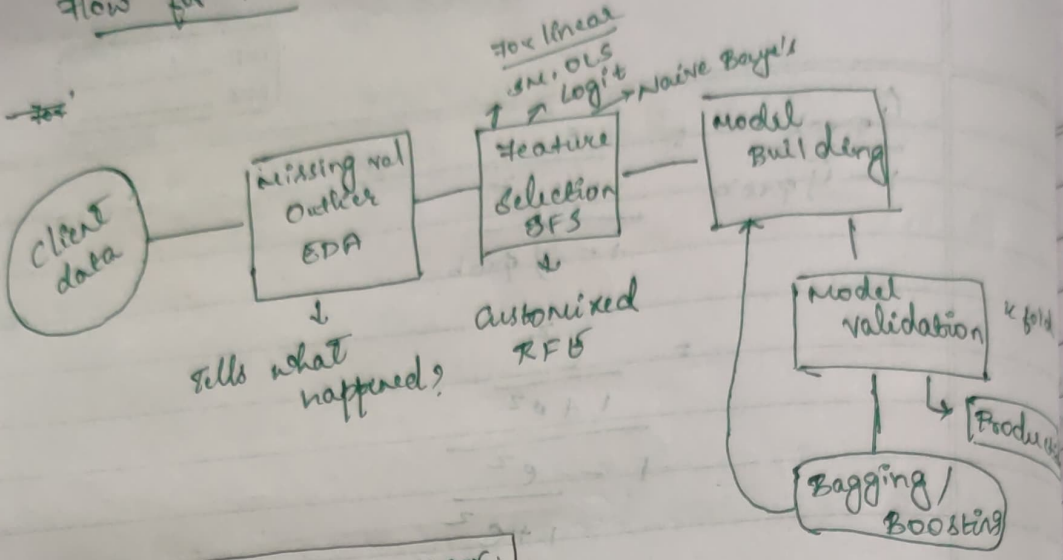
$$\frac{\hat{y}_{prob}}{1-\hat{y}_{prob}} = e^z$$

$$e^{(\beta_0+\beta x)} \Rightarrow \text{odd's.}$$

Raising coeff to $e^{(z)}$ is called odds ratio. In logistic reg, we can't interpret slopes as y.i.s times x as we do in linear regression.

If odds > 1, then odds are significant

Flow for ML



Bagging - ↓es variance error.
Boosting - ↓es bias error. → Can be applied to any model

Drawback of SFS
- Doesn't see pvalue of columns but looks into only R^2
- Drawback of RFE
  User has to specify no of features.

Linear regression - strictly used as Regressor.
Logistic regression - strictly used as classifier
Naive Bayes - strictly used as classifier
} Linear Models

K-NN - Can be used as Regressor or as classifiers - It is Non linear Model

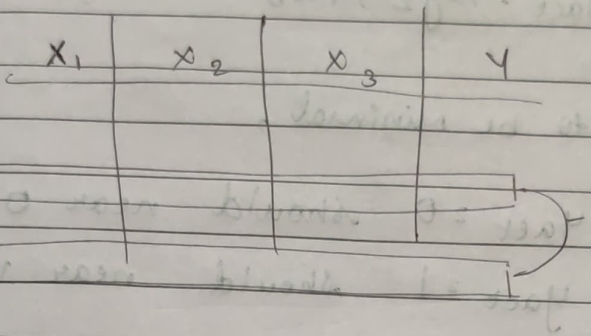
Decision Tree - Regressor & classifier

Random Forest - Regressor & classifier

DATE: / /
PAGE:

Ensemble Models - Non Linear
Bagging & Boosting.

1 sample mean test / 1 sample proportion Test is used to validate the client data.



Shapes:-

X Train
4 × 3

Y Train
4 × 1

X Test
2 × 3

Y Test
2 × 1

Scaling Compulsory Models = KNN & Naive Baye's

Sm. add constant (X)

Constant
1.00
1.00
1.00

This constant is the multiplication factor of β_0 (coeff of β_0)

In Sklearn, model.fit(X, Y)
But in statsmodels, logit(Y, X).fit()

Pseudo $R^2 = 1 - \frac{\text{log likelihood}}{\text{LL Null}}$

↳ practical range of R^2 - 0.2 to 0.6.

loss func of classifiers : log loss (or) entropy

log loss $= -(y_{act} \cdot \log_2 \hat{y}_{prob} + (1 - y_{act}) \cdot \log_2(1 -$

① for log loss to be minimal,

$\hat{y}_{prob}$ for $y_{act} = 0$ should near 0 &
$\hat{y}_{prob}$ for $y_{act} = 1$ should near 1

② Maximum likelihood. → log loss will be minimal

$\hat{y}_{prob}$ = max

$1 - \hat{y}_{prob}$ = max

y_{act}	y_{prob}	
0	0.2	$= 1 - y_{prob} = \hat{y}$
0	0.3	$= 1 - y_{prob} = \hat{y}$
1	0.8	$= y_{prob} = \hat{y}$
1	0.9	$y_{prob} = \hat{y}$

What evaluation metric we choose in validation method matters.

Validation Method

Train / Test split - NO preffered as it is like assessing model with only 1 test

✓ Cross val score

LOOCV - Not preffered as all rec are same & only 1 rec is changed.
(Test data is less & train data is repetitive)

DATE: / /
PAGE:

for Linear Reg:-

```
kf = KFold(n_splits=5, shuffle=True, rand_state)
rmse = cross_val_score(LR, X, Y, cv=kf,
scoring = 'neg_root_mean_sq_error')
```

by default, R^2 → higher the R^2 better.
higher RMSE gives as better.
But, in reality, lowest RMSE is better
↳ Hence 'neg' is given in scoring.

```
np.mean(np.abs(rmse))
np.std(rmse, ddf=1)
```

for Logistic Reg:-

y_{act} $\hat{y}_{pred}$

Y_{act}	Y_{pred}
H	H
H	H
H	H
H	H
H	H
H	H
H	H
H	H
H	H
CVD	H
CVD	H

Confusion Matrix:- (Performance Metric)

ACTUAL \ PREDICTED	H	CVD
H (8)	8	0
CVD (2)	2	0

actual \ Predicted	N	P
N	True Negative — specificity of model	False Positive — Type 1 Error (or) α Error
P	False Negative — Type 2 Error (or) B error	True Positive — Sensitivity of the model

** Type 2 error is dangerous than Type 1

Confusion Matrix is performance metrics
like to RMSE in Linear Reg.
Confusion Matrix - for all classification

Methods to set threshold:

y_{act}	$\hat{y}_{prob}$	$\hat{y}_{class}$ T=0.25	$\hat{y}_{class}$ T=0.7	$\hat{y}_{class}$ T=0.8
H	0.35	1	0	0
H	0.2	0	0	0
H	0.4	1	0	0
CVD	0.68	1	0	0
CVD	0.75	1	1	0
CVD	0.85	1	1	1

	H	CVD
H 3	1	2
CVD 3	0	3

Overall Accuracy = $\frac{4}{6}$

	H	CVD
H 3	3	0
CVD 3	1	2

acc = $\frac{5}{6}$

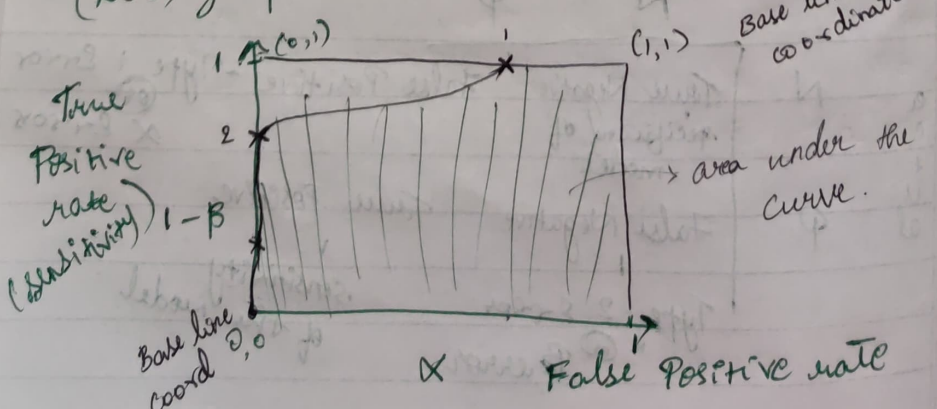
	H	CVD
H 3	3	0
CVD 3	2	1

acc = $\frac{4}{6}$

Accuracy is not sensitive enough to capture the performance - Hence plot.

Receiver operating characteristics (ROC)

(ROC) graph



DATE: / /
PAGE:

Using this graph, we calc Area Under the Curve (AUC)

Map B confusion Matrix on ROC graph.

for 1st one = $\alpha = 2/3 = 66.6\%$
sensitivity = $3/3 = 100\%$

2nd $\alpha = 0\%$
sensitivity = $2/3 = 66.6\%$

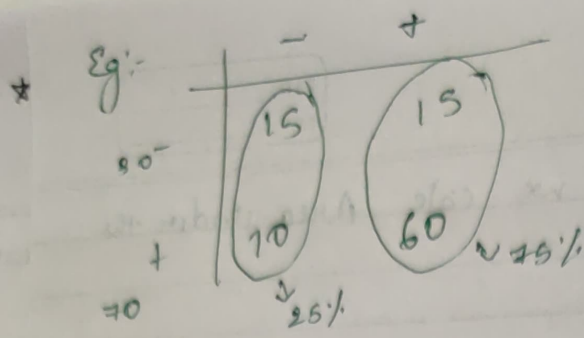
3rd $\alpha = 0\%$
sensitivity = ~~100~~ % $1/3 = 0.33$

Sklearn
↓
model.predict(X)
↳ o/p class label
↓
model.predict_proba(X)
↳ o/p prob ~~of~~ score

sm.logit
↳
model.predict()
↓
o/p prob score
↓
get class label ~~manually~~ manually

kappa score or Reliability score.
↳ Overall accuracy adjusted to Baseline Reference.

$$\text{kappa score} = \frac{OA - P_{chance}}{1 - P_{chance}}$$



OA = 0.75.

Expected by chance

75% of 70 + 25% of 30

52.5% + 7.5% ⇒ 60%

Kappa score = ~~OA~~ = $\frac{0.75 - 0.6}{1 - 0.6}$ = 37.5%.

Multicollinearity.

corr() if < 0.3 OK (No coll)

0.3 to 0.5 moderate coll

> 0.5 High coll

Recall & precision :-

G G G G G S S S S S

G G G G G G S S S S

		G	S		Recall	Precision
5 G	←	5	0	G	100% 5/5	5/6 83%
5 S		1	4	S	80% 4/5	4/4 100%

way can

DATE: / /
PAGE:

$$F1\ score = \frac{2(R \times P)}{R + P}$$

(19) Which threshold makes TPR the highest & FPR the lowest we choose that.
↳ Youden's Index

(20) Sum of α error & β error should be min
↳ Another way to choose threshold.
↳ Choos threshold which gives least total cost

$$\text{Min}(W_1 * FN + W_2 * FP)$$
$$\beta \qquad \alpha$$

(5) Simply choose Mean.

3 ways ↑ to find threshold