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 inter relations.

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 $\quad -(y - \hat{y})$

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

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

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

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 (value) 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:

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

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

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

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

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

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

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

$< 0.5 \Rightarrow$ class = 0

Concept of Odd's

Mutually Exclusive → Union | Addition.

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

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

$P_{SE} = 0.2$

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

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

Covishield

$P_{SE} = 0.1$

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

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

$$\frac{O_{CX}(P_{SE})}{O_{CS}(P_{SE})} = 2.3$$

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

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

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

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

DATE: / /
PAGE:

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

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

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

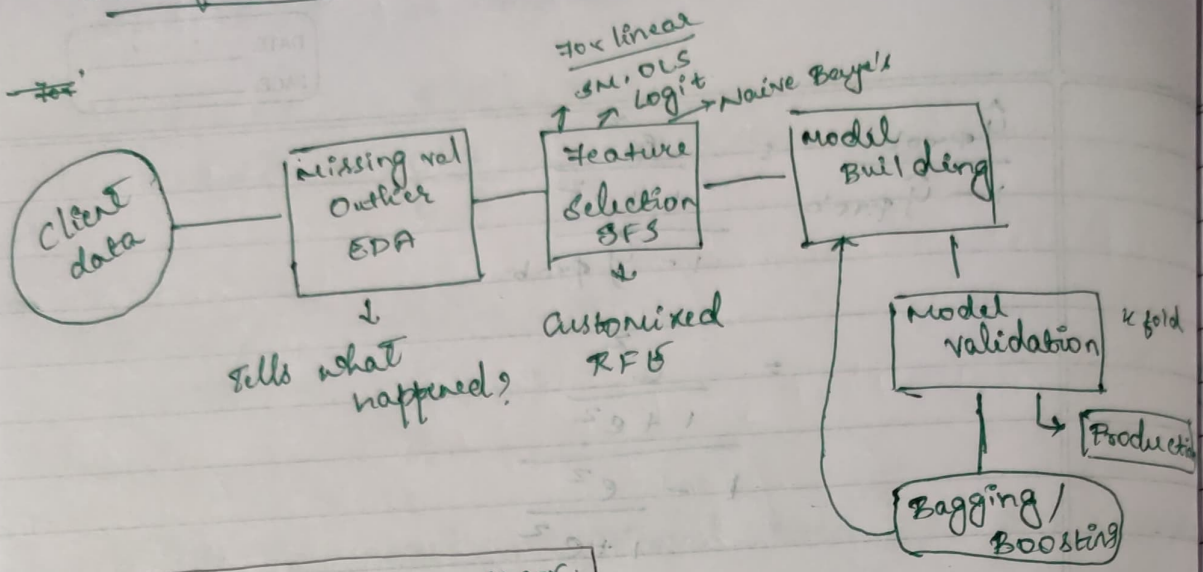
$$\frac{\hat{y}_{prob}}{1-\hat{y}_{prob}} = e^z$$

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

Raising coeff to $e^{(z)}$ is called odds ratio. In logistic reg, we cant interpret slopes as y.s times x as we do in linear regression.

If odds > 1, then odds are significant

Flow for ML



Bagging - ↓ss variance error.
Boosting - ↓ss 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

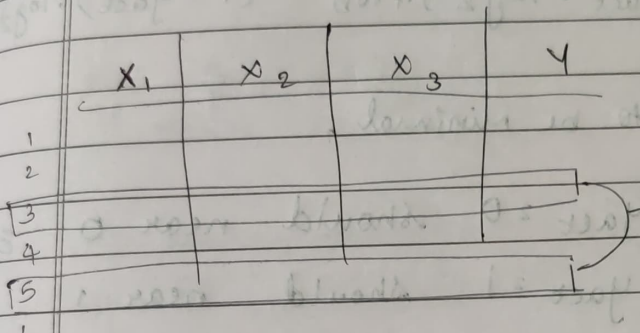
Decision Tree - Regressor & classifier

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

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

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) β Error	True Positive — Sensitivity of the model

** Type 2 Error is dangerous than Type 1.

Confusion Matrix is performance metrics
w.r.t to RMSE in Linear Reg.
Confusion Matrix - for all classification models

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
0 -	H	0.35	1	0	0
	H	0.2	0	0	0
	H	0.4	1	0	0
1 -	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 = $\frac{5}{6}$

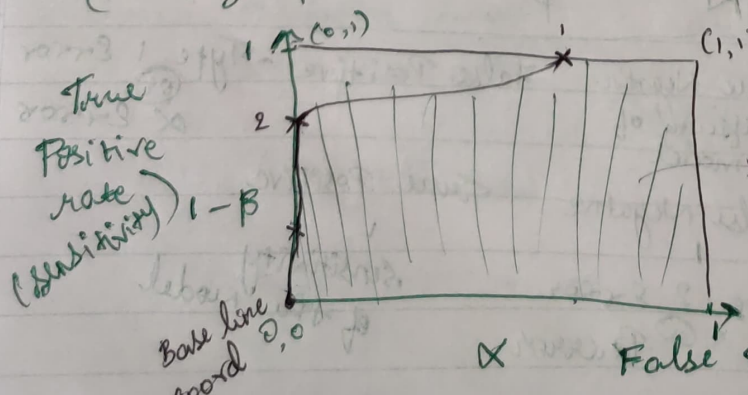
	H	CVD
H 3	3	0
CVD 3	2	1

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 3 Confusion Matrix or 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 = ~~33~~ % $1/3 = 0.33$

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

↓

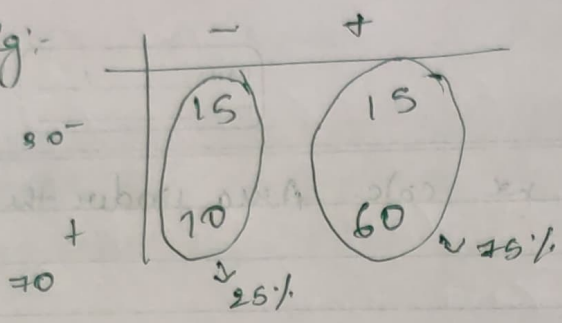
get class label ~~use~~ manually.

kappa score or Reliability score.

↳ Overall accuracy adjusted to Baseline Reference.

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

Eg:-



OA = 0.75

Expected by chance

75% of 70 + 25% of 30

52.5% + 7.5% ⇒ 60%

Kappa score = $\frac{0.75 - 0.6}{1 - 0.6}$ = 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	was CM	Recall	Precision
5 G	G	5	0	← G	100% 5/5	5/6 83%
5 S	S	1	4	S	80% 4/5	4/4 100%

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

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

(3) Simply choose Mean.

3 ways ↑ to find threshold

NAIVE BAYE'S.

Mutually Exclusive Variables

$P(H \text{ or } T) = P(H) + P(T)$
$= 0.5 + 0.5 = 1$

$P(K \text{ or } Q) = P(K) + P(Q)$
$= \frac{4}{52} + \frac{4}{52} = \frac{8}{52}$.

$P(A \cap B) = 0$

Mutually Non Exclusive Variables

$P(A \cap B) \neq 0$

$P(K \text{ or } \Delta) = P(K) + P(\Delta) - P(K \cap \Delta)$

$= \frac{4}{52} + \frac{13}{52} - \frac{1}{52}$

Mutually Independent Variables

$P(A \text{ and } B) = P(A) \cdot P(B)$

Conditional Probability

Mutually Non Ind Var

$P(A \text{ and } B) = P(A) \cdot P\left(\frac{B}{A}\right)$

A is affecting B

Or - Union
and - Intersection

Baye's Theorem.

$$P(B)\ P(A/B) = P(A)\ P(B/A)$$

$$P(A/B) = \frac{P(A)\ P(B/A)}{P(B)}$$

Naive Baye's classifier.

Color Model Trans Booking Y/N.

$$P(\text{Booking} = \text{'Yes'} / \text{Red, Asta, MT}) = \frac{P(\text{Red/yes}) * P(\text{Asta/yes}) * P(\text{MT/yes})}{P(\text{Red}) * P(\text{Asta}) * P(\text{MT})}$$

$$P(\text{Booking} = \text{'No'} / \text{Red, Asta, MT}) = \frac{P(\text{Red/No}) * P(\text{Asta/No}) * P(\text{MT/No})}{P(\text{Red}) * P(\text{Asta}) * P(\text{MT})}$$

Classifier classifies based on probability which is higher (Denominator is same & doesnt matter)

For Normal distribution

$$\frac{1}{\sigma\sqrt{2\cdot\pi}}\, e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Never scale y.

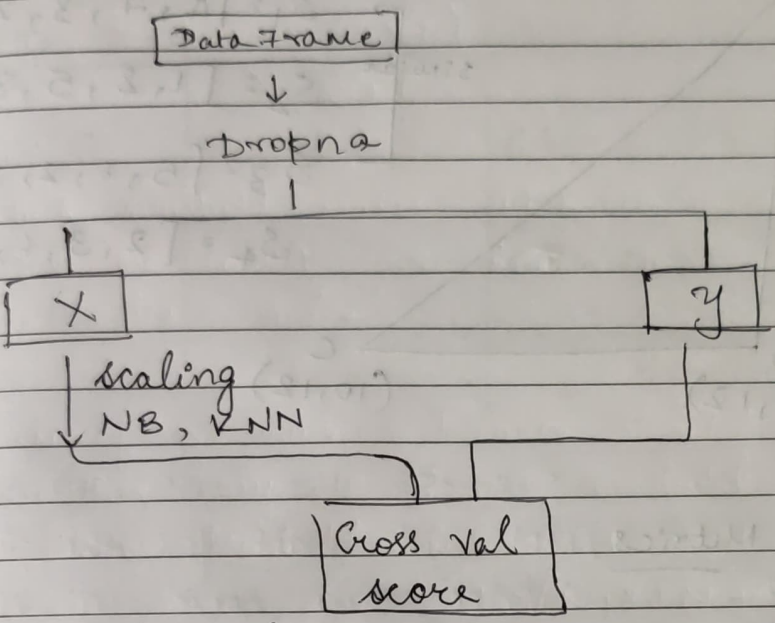
$z = \frac{x - \bar{x}}{\sigma}$ 1.2×3.2

DATE: / /

PAGE:

38.4
25.30
3.82
29.12

* classifiers, bias Error is the balance or 100%. — auc-roc / wt



cross_val_score (model, X_scaled, y, cv = kfold, score =
bi class = 'roc-auc'
multi class = 'f1-~~score~~ weight'
log reg = 'neg-mean-square...'
lin reg = r square

No feature selection in cross val score happens

→ Bias Error = 4.190 Target price in $
Var Error = 0.481

Inference :-

~~0.481 ± 4.190~~

4.190 ± 0.481 ×1000

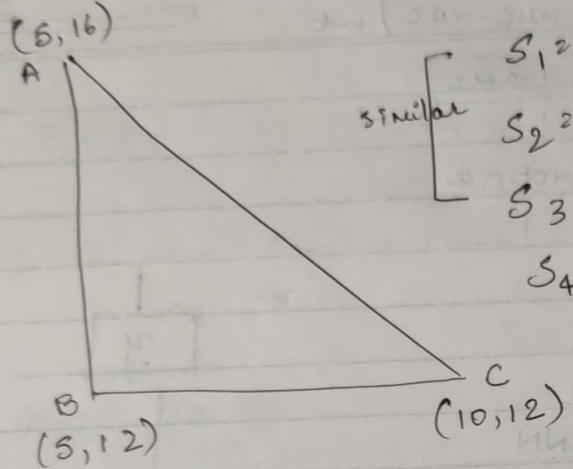
4190 $ ± ~~0~~ 481 $

Pred ~~Avg~~ car price is

Avg difference b/w the actual price & predicted price will lie b/w 3709 & 4608.

K - Nearest Neighbour (KNN)

(Lazziest Alg 😊)



similar

$S_1 = [4, 7, 3, 2]$

$S_2 = [1, 2, 5, 8]$

$S_3 = [5, 6, 2, 3]$

$S_4 = [2, 3, 6, 9]$

similar

Distance Metrics

1) Euclidian Distance [can travel diagonally]

Distance b/w 2 points A & C $= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

2) Manhattan Distance [cant travel diagonally]

Has to travel straight

$|x_1 - x_2| + |y_1 - y_2|$

3) Cosine Distance [Commonly used in Recommendation System].

First compute Cosine Similarity Score. (CS)

Cosine distance = $(1 - CS)$

$$CS = \frac{\vec{A} \cdot \vec{C}}{\|\vec{A}\| \cdot \|\vec{C}\|}$$

$$CS = \frac{(5,16) \cdot (10,12)}{\|(5,16)\| \cdot \|(10,12)\|} = \frac{50+192}{\sqrt{5^2+16^2} \cdot \sqrt{10^2+12^2}}$$

DATE: / /
PAGE:

	ED	MD	CD (1-CS)
(closer) S1-S3	2.0	4	CS = 0.97 CD = 0.03
(far) S1-S2	8.6	16	CS = 0.57 CD = 0.428

For (S_1, S_3) & (S_1, S_2)

$$S_1S_2 = \sqrt{(4-1)^2+(7-2)^2+(3-5)^2+(2-8)^2}$$

$$= \sqrt{3^2+5^2+2^2+6^2} = 8.6.$$

$$S_1S_3 = \sqrt{(4-5)^2+(7-6)^2+(3-2)^2+(2-3)^2}$$

$$= 2.$$

S_1-S_3 $$\frac{20*4+2+6+16}{\sqrt{4^2+7^2+3^2+2^2}\cdot\sqrt{5^2+6^2+2^2+3^2}} = \frac{84}{}$$

S_1S_2 $$\frac{4+14+15+16}{\sqrt{4^2+7^2+3^2+2^2}\cdot\sqrt{1^2+2^2+5^2+8^2}} = \frac{49}{}$$

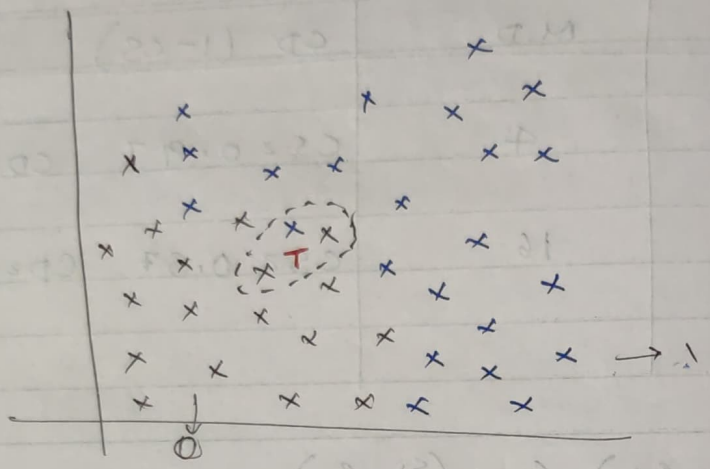
Default mode of op^n / Unwaited or Uniform mode

Hyper Parameter

Working of KNN:

n = 3.

optimal n-neighbour
2 - 5% of datasize



Blue - 1
Black - 0
T - Test rec

KNN will wait for the Test records once they come in it will place it on value and calculate the distance from that test value to all the points and arranges it in ascending order & picks the first 3 and predicts it to be the max class of 3 (here it is black-0)

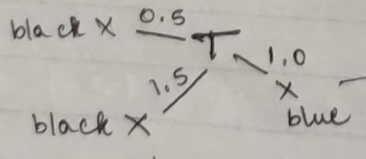
2nd Hyper parameter

k Nearest Neighbor (n-neighbor, weights = 'distance')

by default, weights = 'uniform'.

~~1~~ 2) Weighted Voting

$WV = \frac{1}{0.5^2} + \frac{1}{1.5^2}$



Weighted Voting

$WV_{blue} = \frac{1}{dist^2} = \frac{1}{1^2} = 1$

$WV_{black} = \frac{1}{1.5^2} + \frac{1}{0.5^2} = 4.44$

KNN cannot be Boosted

DATE: / /
PAGE:

model predicts the one which has NV more. Weighted mode - makes the model overfit

→ Still we prefer this mode only. as we can bagg it later. But never take risk of underfitting by doing unweighted / uniform.

→ More refined results rather than going with the crowd.

Cans also be used as Regressor.
Both weighted & Unweighted can be done.
Avg of 3 as predicted.

180 mg
190 mg
200 mg

If unweighted avg glu lvl = 190 mg,
weighted avg will be < 190.

$$\frac{1}{0.5^2} * 180 + \frac{1}{1^2} * 190 + \frac{1}{1.5^2} * 200 \quad < 190$$

New mean will get attracted to the one with least distance.

Cross tab.

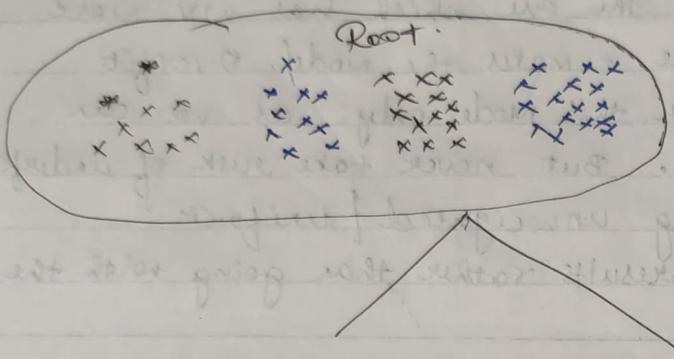
Last in Q16 Day 3 Reclass

	Bias Error	Var Error.
NB	11.3	0.0115
KNN	6.4 (100 - 93.6) (100 - KNN mean)	0.0074 direct SD is taken

As BE & VE both are low in KNN, KNN is a better model

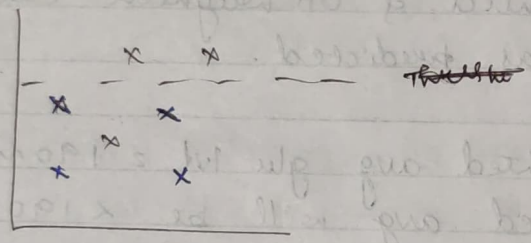
Complexity

Decision Tree working.



Forms Axis line (horizontal | vertical) but not inclined

split -



Which feature, what threshold can get best

Entropy or Ginni Score

- Measure of Uncertainity

At root level, before splitting.

-(3/7 * np.log2(3/7) + (4/7) * np.log2(4/7)) - entropy

Entropy range - 0 to 1

Ginni score - 0 to 0.5

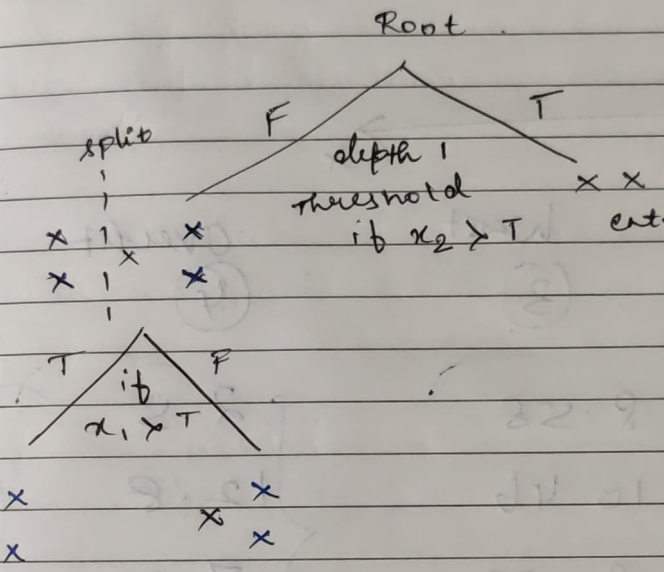
1 - ((3/7) **2 + (4/7) **2) - Ginni score

Aim of DT is to minimize uncertainity

weighted entropy = entropy x $\frac{\text{how many points}}{\text{Total points}}$

DATE: / /
PAGE:

Difference b/w root Entropy and weighted entropy of depth 1 is known as information Gain.



fully grounded decision Tree

Entropy=0

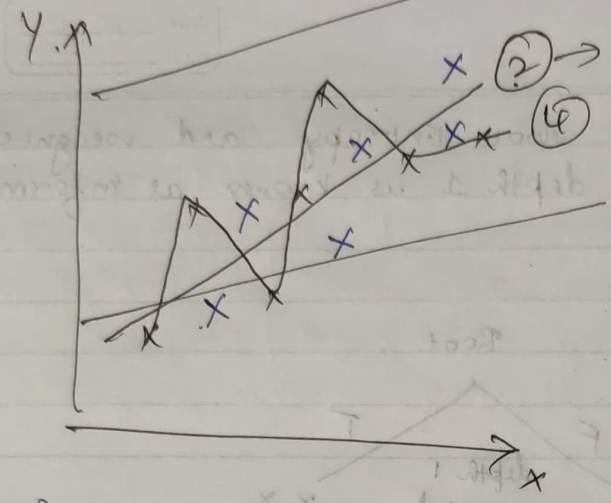
DT as Regressor.

Metric is MSE.

At root level, MSE = total mean.

DT has highest bias error & Variance error.
Hence, we cut off last few leaves (pruning)

Over pruning results in underfitting.



Best ②

4.36
4.24
4.32
4.0
4.56

4.4

low BE
low VE

Worst ③

8.58
10.46
9.58
8.72
8.86

8.8

High BE
low VE

Overfit ④

2.5
12.8
7.5
10.76
3.4

4.8

High VE
low BE

DATE: / /
PAGE:

	Y1	Y2	Y3	Y4	Y5	
S1	90	92	86	85	92	→ Vet,
S2	95	70	58	98	60	→ Ve
S3	35	34	40	45	42	

↓ Vet. BE↑

40

BE 60%.

Ve

$7500

$3400 ✓