Featurization and Feature engineering

Introduction :-

Text data :- Bow, tfidf, avgw2v, tfidf w2v

also sequence data

various featurization of text data.

Categorical data :-
- one hot encoding
- mean - response rate,
- domain Specific.

Time - Series :-
- heart rate
- ecommerce
- Speech / audio
- Stock market price

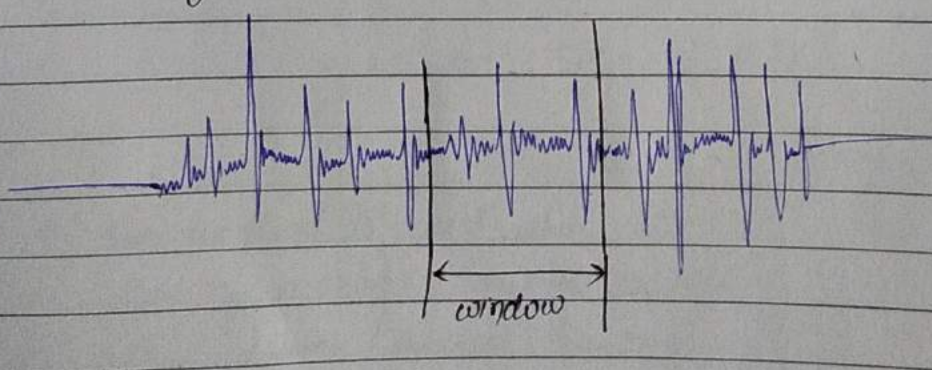
Youtube: Programming Cradle

Image data :- Face detection, face - recognition
X-rays, MRI scans, video

Img + time Series.

Graph data :- recommend a friend on FB

Moving window for Time Series Data

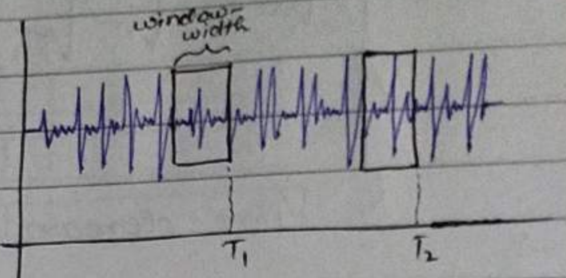


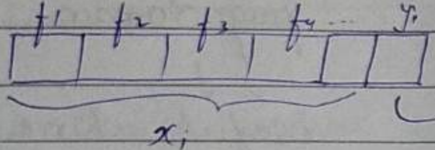
- mean, std-dev
- median, Quantile
- max, min
- local maxima
- Zero crossing / mean crossing

how many times wave curve crosses mean line

∴ window width depends on data.

Steps of featurization :-

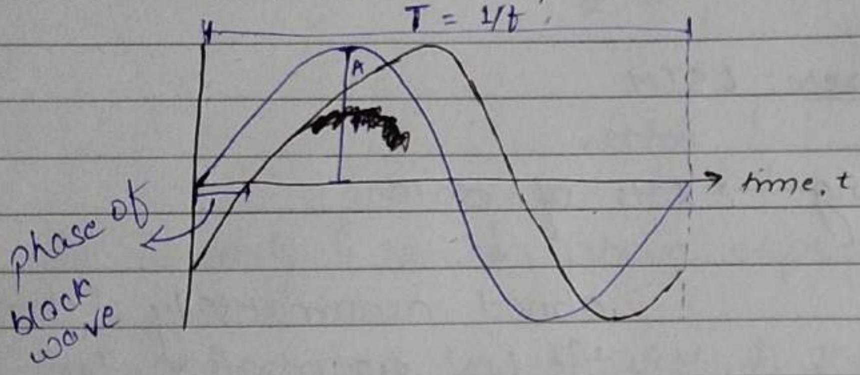


- Decide window width
- Features that are useful
- 

↳ boolean : heart attack in next 10 min or not.

Fourier decomposition for Time Series Data :-

↳ Method to represent Time-Series.



A : amplitude
T : period
f : frequency.

eg: if 1 osc. per second → 1 Hz
2 osc. per Second → 2 Hz.

heart rate :- 60-100 beats per minute

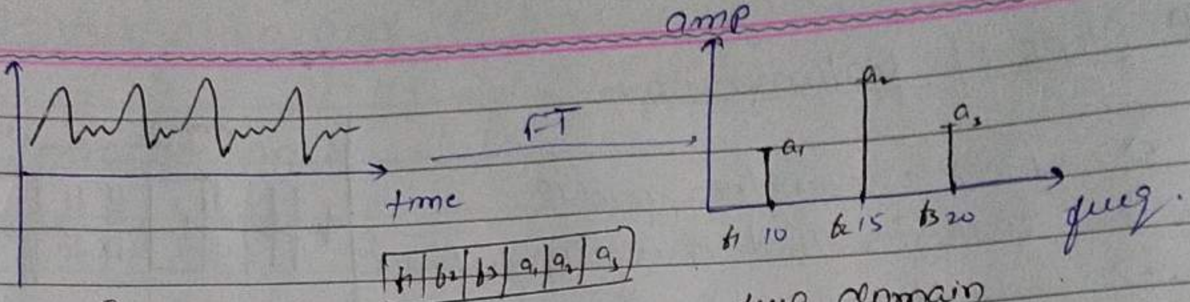
$\Downarrow$ $\underbrace{\text{beats}}_{\text{osc.}}$

1-1.6 per Second. ⟹ 1 Hz to 1.6 Hz

Fourier Decomposition / Transform :-

A composite / complex wave can be represented as sum of various Sine ways and convert it to freq vs Amp. plot.

Refer wikipedia FFT-Time-frequency-view.png



Time - domain

freq. domain ↓ useful whenever repeating pattern.

NOTE :- Whenever repeating pattern is there frequency is important.

⊛ Deep Learning features : LSTM

Deep - Learning ⟶ Lots of data ↓ almost automatically learn the best featurization for data

↓
{ Time - Series
Text - data
Image - data }

Deep Learnt features ⟶ best features today.

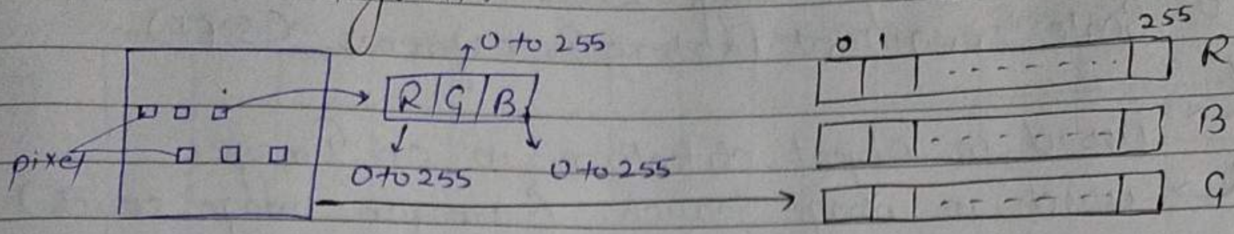
⊛ Image Histograms :-

Youtube: Programming Cradle

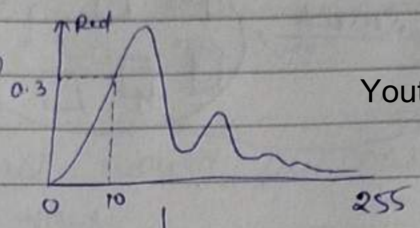
Images :- faces, object, scans, x-rays, autonomous car

Two popular Types of histograms:-
↳ Color histogram
↳ Edge histogram.
} basic and rudimentary

⊛ Color histogram :-



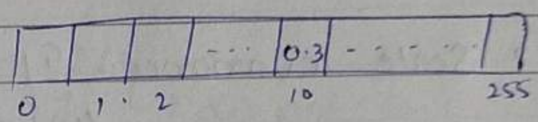
① Collect Red values for each pixel
↓
histogram



Youtube: Programming Cradle

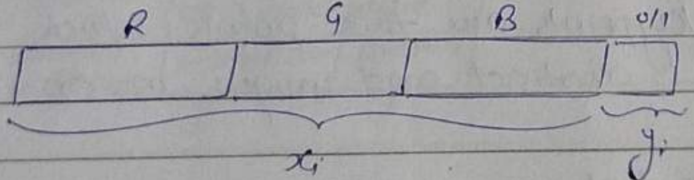
↓

vector



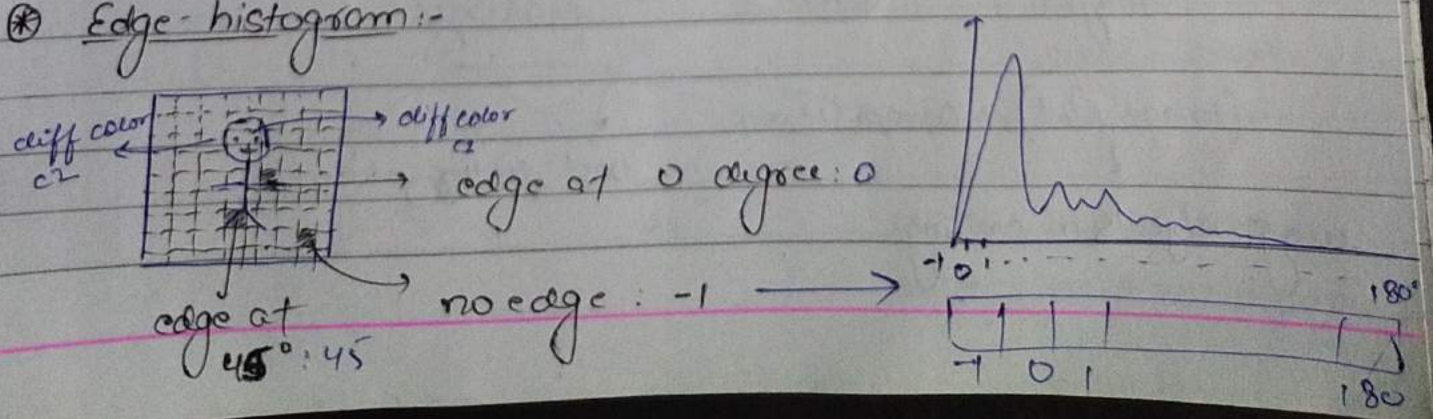
② Similarly I can do above steps for Green and Blue.

Example :- We want to detect if sky is there in the image or not.



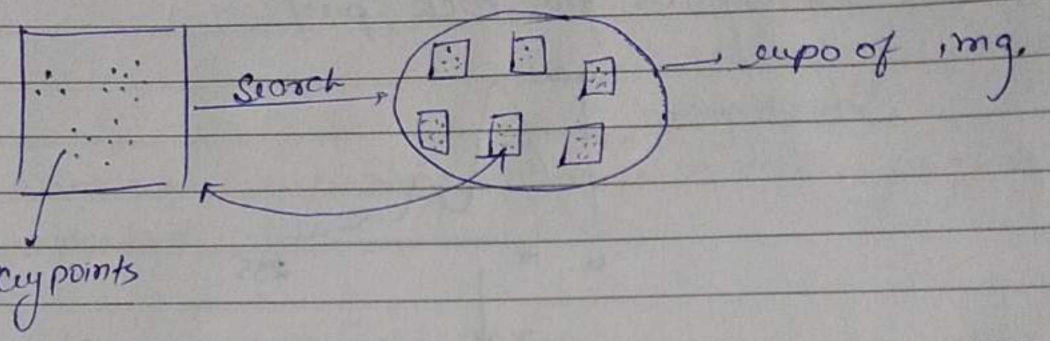
if sky present → B will have more values → 1 = y_i
if sky not present → B will have less values → 0 = y_i

⊛ Edge-histogram :-



⊛ Keypoints :- SIFT for image Data
Scale Invariant Feature Transforms (SIFT)

↳ used in object detection in an Image.
↳ used in image search. (amazon image search)



- Scale invariance :- If image is bigger or smaller features donot change much.
- Rotation invariance :- If image is Rotated slightly features do not change much (may change / not work when too much rotation)

Youtube: Programming Cradle

- Keypoint :- Keypoints are those points which are very distinct and unique in an image

⊛ Deep learning features : CNN :-

Time - Series → LSTM → featurize

Images → CNN → featurize.
↓
best featurization

(image Net) → competition
↓
recogn objects in an img

⊛ Relational Data :-

eg.

custid	cust pin code

cust table

custid	pid	time

cust viewing/ visitation data

custid	pid	Time

purchase data

prid	prtype.

product data

Relational DBs :- Mysql, Oracle, SQL server

⊛ Task : predict if a cust would purchase a product in the next 7 days.

(Cust id , product id) → 1/0

Given

1 → buy

0 → do not buy

f1 f2 f3 ... y_i

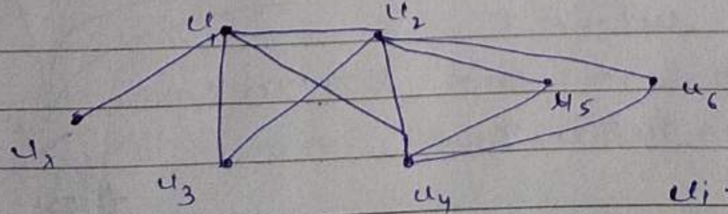
f1 : no. of times the cust id visited the product page in last 24 hrs

f2 : no. of times custid visited any product type same as pid

pin code (certain pincodes have richer people)

⊛ Graph Data :-

eg:-



social - graph

u_i → vertex (user)

edge → relation (friendship)

Youtube: Programming Cradle

* Task :- Recommend new friends for a user u_i

user	suggestions
u_4	$\{u_3, u_7\}$

f_1 :- no. of mutual friends
f_2 :- no. of paths b/w 2 users

* Indicator variables :-

eg:- ① 'height' as a feature
- ↳ keep it as is $h \in \mathbb{R}$ → h → real valued feature
- ↳ convert it to indicator variable
  - ↳ if $h > 150 \longrightarrow 1$, $h < 150 \longrightarrow 0$ } binary indicator feature.

eg:- ② categorical feature :- Country.

indicator variables {

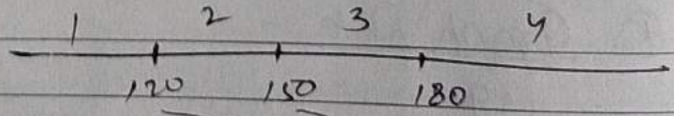
```
if Country == 'India' or country == 'USA' :
        return 1
else
        return 0
```

* Feature binning :-

↳ Extension to indicator var.

eg:- hight $\in \mathbb{R}$

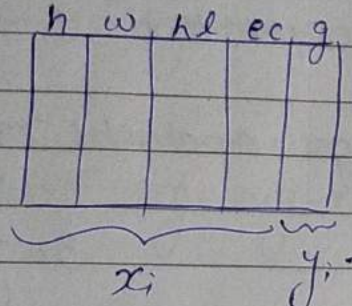
```
if h < 120
    return 1
if h < 150 and h >= 120 cm.
    return 2
:
so on
```



these thresholds are problem specific.

⊛ Task :- Predict gender given :- h, w, hair length, eye color

D →

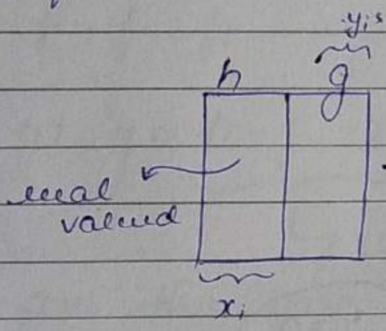


y_i → 1 male, 0 female

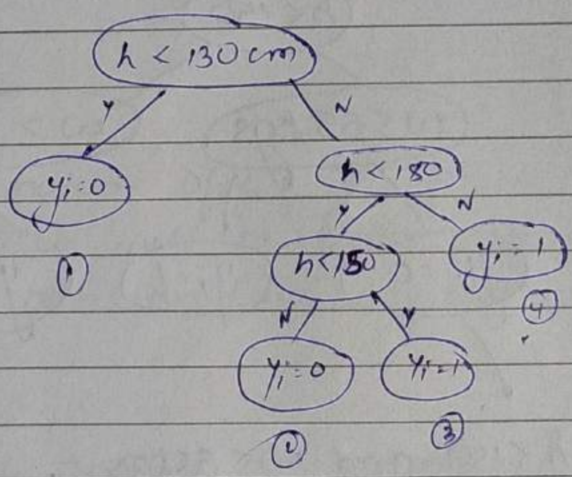
height → binning.

Challenge :- binning of 'h' using D (without domain knowledge)

One of the sol^n :-



Simple DT →



if h < 130
  return 1

if h ≥ 130 and h ≤ 150
  return 2

if h ≥ 150 cm and h < 180 cm
  return 3

if h ≥ 180 cm
  return 4

binned real-valued features using y_i's and feature itself using DT

⊛ Interaction variables :-

Task $\underbrace{h, w, hl, ec}_{x_i}$ → $\underbrace{\text{gender}}_{y_i}$

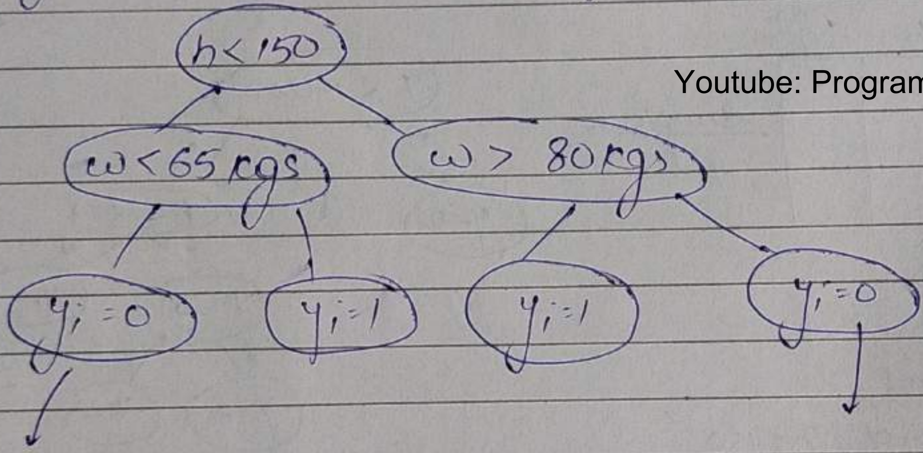
eg :- ① (h < 150 cm) and (w < 60 kgs) → logical 2 way interaction feature eg> ② $h * w \rightarrow f_2$ ∴ operation depends on use case, can be any operator $+, -, *, /$ ... etc.

eg> ③ $h < 150$ cm and $w < 65$ kgs and $hc > 5$cm → 3 way interaction

Q) Given a task, how to find good interaction features?

Sol^n DT

eg $x_i = h, w, hc$, y_i = gender



Youtube: Programming Cradle

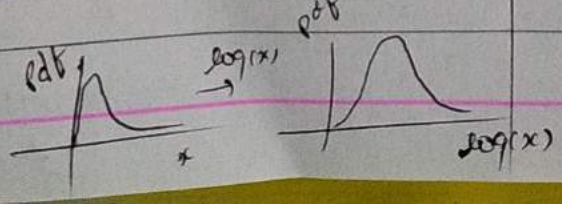
$h < 150$ and $w < 65$kgs

$h \geq 150$ cm and $w \leq 80$ kgs.

⊛ Mathematical Transforms :-

$x \rightarrow$ Single feature

Math Transforms
- $\log(x)$, e^x
- sqrt(x), $\sqrt[3]{x}$
- x^2, x^3 ... polynomial
- $\sin(x)$, $\cos(x)$, $\tan(x)$



Q. What is the best transform?

A) Problem specific.

eg) if $x \rightarrow$ ~~power law dist.~~ log normal dist.

↓

$\log(x) \rightarrow$ best transform bcoz it will make the dist. as normal dist.

Model specific featurizations :-

eg:- let $f_1 \rightarrow$ log normal $\Rightarrow \log(f_1) \rightarrow$ Gauss. dist.

we want to use logistic regression and we know that log-reg $\rightarrow$ Gauss NB. $\rightarrow$ bcoz of assump^n we can not use f_1 directly hence we will trans-form using $\log(f_1)$
can be used for logistic reg^n

assumption:-
· feature is normally dist.

eg2 f_1, f_2, f_3 , $y \in \mathbb{R}$

let $y \approx \underbrace{f_1 - f_2 + 2f_3}$
↳ linear combination of f_i's

→ for this case linear models will be better option like linear regression. and decision tree may not be best option.

eg. 3 $y \rightarrow$ interactions of f_1 and f_2 ← domain knowl.
h,w → gender ∴ DT / RF / GBDT best opt^n.

- text data : (BOW) ⟶ linear models are better most of the times. as they have high dim. and linearly separation is easy in higher dim.

Feature Orthogonality:-

~~*> The more the feat~~
*> The more diff / orthogonal the features are, the better would your model be.

eg:- $f_1, f_2, f_3 \longrightarrow y$.

let if $\underbrace{f_1 \xrightarrow{corr} y \quad f_2 \xrightarrow{corr} y \quad f_3 \xrightarrow{corr} y}_{\text{very good.}}$

but let

$f_1 \xrightarrow[corr]{highly} f_2, \quad f_1 \xrightarrow[corr]{high} f_3, \quad f_2 \xrightarrow[corr]{high} f_3$

→ because of high correlation among features effect will be low.

eg:- $f_1, f_2, f_3 \longrightarrow y$.

let if $f_1 \xrightarrow{corr} y, \quad f_2 \xrightarrow{corr} y \quad f_3 \xrightarrow{corr} y$

and $f_1 \xrightarrow[corr]{not} f_2 \qquad f_1 \xrightarrow[corr]{not} f_3$

→ in this case effect overall will be good.

NOTE :- while creating new feature we should try (ideally) new feature is highly corr with dependent ~~feature~~ var (y) and not corr with other independent ~~features~~ var (f_i)

Q) How do I design a new feature ~~f_g~~ f_4 s.t. $f_4 \xrightarrow{corr} y_i$ and less corr with $f_1\, f_2\, f_3$?

1 of the soln:-
or idea:- $\forall x_i$ find $error_i = y_i - \tilde{y}_i$ $\quad f_4 \longrightarrow error_i$

$f_1\, f_2\, f_3\, f_4 \longrightarrow y_i$

idea of GBDT :- GBDT → model 1 → errors → model 2 → errors
errors → new model ... so on.

⊛ Domain Specific featurizations:-

eg.> heart attack → ECG data

using this we can create new features. ← { important to research and study existing featurizations doctors/specialists.

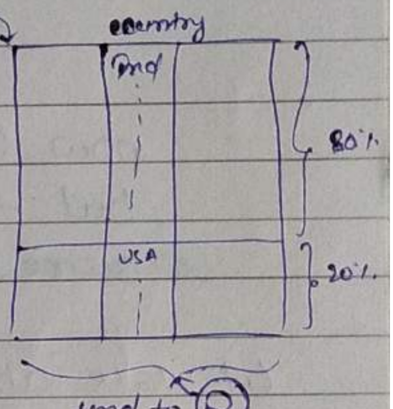
⊛ Feature slicing:-

Youtube: Programming Cradle

ec → eye color

eg.> $\underbrace{h, w, hL, ec, country}_{x_i}$ ⟶ $\underbrace{\text{gender.}}_{y_i}$

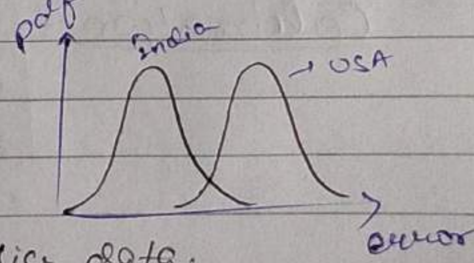
country: India, USA



- In this case where we have more people / datapoint from India model will perform better and for USA, model tend to perform poor.

Q) How to detect if more people are from specific country and less from other?

A) $D \xrightarrow{train} M \xrightarrow{test}$ $error_{india}$, $error_{USA}$ $\xrightarrow{plot}$



→ To get rid of above problem we can slice data.

$D \rightarrow D_{ind} \rightarrow M_1$

$D \rightarrow D_{USA} \rightarrow M_2$

→ Make ~~diff~~ separate models for each D_{ind} and D_{USA}

- Slicing the data based on feature's category. when
  1. category1 and category2 are diff^n
  2. Sufficient no. of pts for each slice of data.

⊛ Calibration of Models: Need for calibration.

eg:- let 2 class classf^n $y_i \in \{0,1\}$, $D_{train} = \{x_i, y_i\}$ $\xrightarrow{train}$ $\boxed{\text{Model } f(x)}$

now given $x_q \Rightarrow f(x_q) = y_q$ (1 or 0)

now I want to understand $p(y_q = 1 / x_q, f(x))$
but the o/p of f^n may or not be a prob. score

let talk about NB, we calculate $p(y_q=1/x_q)$ and $p(y_q=0/x_q)$ which is proportional to, prior * likelihood but not equal.
So, $f(x_q) = y_q$ and y_q may or maynot be the actual $p(y_q = 1 / x_q)$

Calibration:- Calibration is comparision of the actual output and the expected output given by a system.
Youtube: Programming Cradle
We calibrate our model when the probability estimate of a data point belonging to a class is very important.

eg:- let 2-class

$$\text{log-loss} = \frac{-1}{n} \sum_{i=1}^{n} y_i \log p_i + (1-y_i) \log(1-p_i)$$

here we use actual p_i, ~~hence~~ if p_i's are wrong log-loss will be wrong. hence we need calibr^n.

$x_q \longrightarrow \underbrace{f(x_q) = y_q}_{\text{Model}} \longrightarrow y_q \longrightarrow$ calibr^n model $\longrightarrow P(y_q = 1/x_q)$

⊛ Calibration Plots :-

$D_{Train} \longrightarrow f^n$

probability of given data point belonging to a specific class.

① $D_{cv} \longrightarrow f^n \longrightarrow \hat{y}_i$:)

x_i	$\hat{y}_i$ ($f(x_i)$)	y_i (true class label)
x_1	$\hat{y}_1$	y_1
x_2	$\hat{y}_2$	y_2
⋮	⋮	⋮

② Sort this table in ↑ order of $\hat{y}_i = f(x_i)$

$\hat{y}_1 \le \hat{y}_2 \le \hat{y}_3 \le \dots$

③ Break table obtained in second step in chunks.

Chunks of Size 100

let D_{cv} = 10k pts

x_1	$\hat{y}_1$	y_1	avg $\hat{y}_i = \frac{1}{m}\sum_{i=1}^{m} \hat{y}_i = \hat{y}^{(1)}$; avg $y_i = \frac{1}{m}\sum_{i=1}^{m} y_i = y^{(1)}$
⋮	⋮	⋮	
x_{100}	$\hat{y}_{100}$	y_{100}	
x_{101}	$\hat{y}_{101}$	y_{101}	$\hat{y}^{(2)}$, $y^{(2)}$
⋮	⋮	⋮	
x_{200}	$\hat{y}_{200}$	y_{200}	
x_{201}	$\hat{y}_{201}$	y_{201}	$\hat{y}^{(3)}$, $y^{(3)}$
⋮	⋮	⋮	
x_{300}	$\hat{y}_{300}$	y_{300}	

:> no. of pts in each chunks = 100

:> no of chunks $= \frac{10K}{100} = 100$

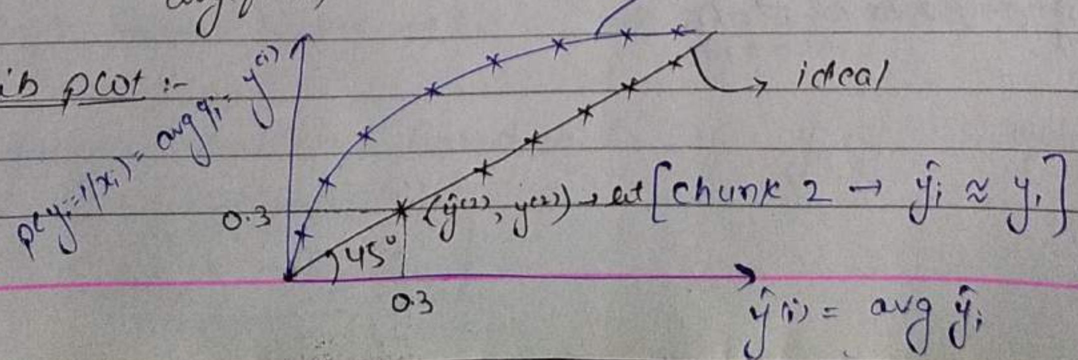
Youtube: Programming Cradle

$D_{calib} = \{\hat{y}^{(i)}, y^{(i)}\}$

$\hat{y}^{(i)} \rightarrow$ avg$(f(x_i))$

$y^{(i)} \rightarrow P(y_i = 1/x_i)$

⊛ Calib plot :-



⊛ Platt's Scaling / Sigmoidal Calibration :-

$$P(y_q = 1 / x_q) = \frac{1}{1 + \exp(A f(x_q) + B)}$$

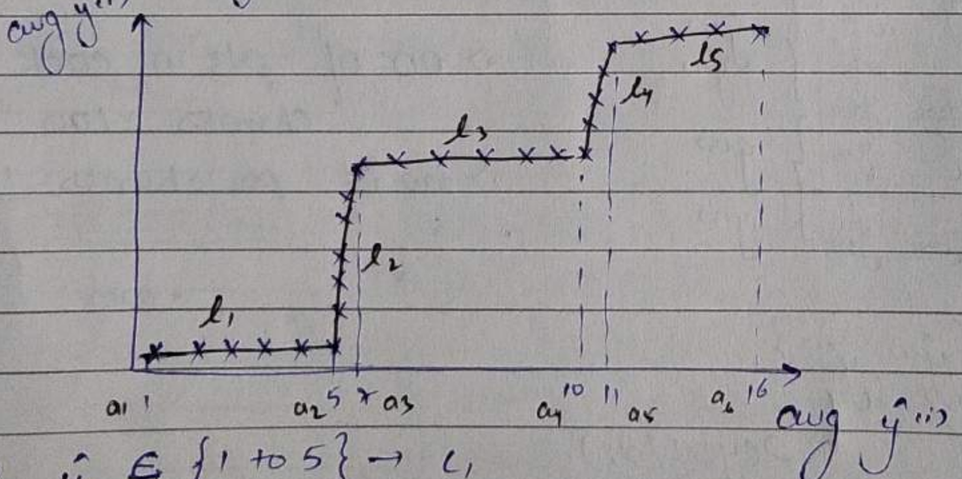
Q. How to find A and B?

A) Replace $f(x_q) \rightarrow \hat{y}^{(i)}$ and $P(y_q = 1/x_q) \rightarrow y^{(i)}$ and then find Best A and B by solving optimization problem. (there are various formulat? of this opt^m problem)

NOTE :- Platt's calibration works iff calib. plot looks like sigmoidal f^n

⊛ Isotonic Regression :-

↳ Works even if calib plot doesnot look like a sigmoidal f^n.



isotonic reg. :- piece wise linear model

if $\hat{y}_q \in \{1 \text{ to } 5\} \rightarrow l_1$

$\hat{y}_q \in \{7 \text{ to } 10\} \rightarrow l_2$

$a_1, a_2, a_3, a_4, \dots$

$\underbrace{a_1, a_2}_{l_1}$, $\underbrace{a_2, a_3}_{l_2}$, $\underbrace{a_3, a_4}_{l_3}$

(m_1, b_1) (m_2, b_2) (m_3, b_3)

m → slop
b → intercept

⇒ optimization problem :- (find threshold a_i) $\xrightarrow{then}$
- find optimal m_i and b_i for each line
- Also there shouldn't be to many line (just enough no. of lines)

• NOTE :- In isotonic reg, need many more points / data in D_{calib} than with platt scaling (A,B) because in isotonic reg we have to find many more parameters $(a_i), (l_i)$

- Large Data → Large CV → Large Calib Data → Isotonic reg.
- Small Data → Small CV → Small D_{calib} → Platt scaling.

⊛ Random Sampling Consensus (RANSAC) [modeling in the presence of outliers]

↳ Can we build a robust model in the presence of outliers?

Let our model is Lr. reg. $\xrightarrow[RANSAC]{without}$ our hyperplane might get shifted towards dominating outliers present in the data. hence model is not robust.

⊛ RANSAC steps :-

① D_0 = $\underbrace{\text{random sample from}(D_{Tr})}_{\text{let } 50\% \text{ of } D_{Tr}}$ → This will also ↓ no. ~~chance~~ of outliers by ≈ 50%

↳ model Lr reg. → L_0

② Compute outliers based on L_0 if $y_i - \hat{y}_i$ is high → outlier

O_0 = {outlier pts}

$$\boxed{D'_{Tr} = D_{Tr} - O_0}$$

③ $D'_{Tr} \xrightarrow[\text{sampling}]{\text{random}} D_1 \xrightarrow{\text{modeling}} \text{l.r. reg}$

$\text{l.r. reg} \xrightarrow{\text{remove outliers } (O_1)} D^2_{Tr} = D'_{Tr} - O_1$

$\text{l.r. reg} \xleftarrow{\text{modeling}} D_2 \xleftarrow[\text{sampling}]{\text{random}} D^2_{Tr} = D'_{Tr} - O_1$

$\text{l.r. reg} \xrightarrow{\text{remove outliers } (O_2)}$

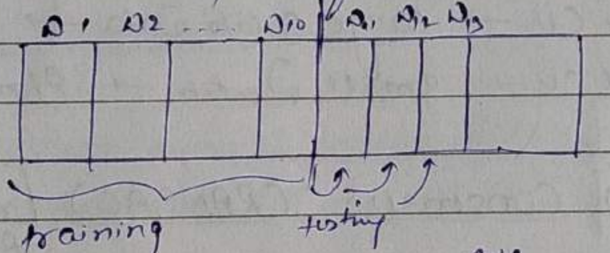
$D^2_{Tr} - O_2 = D^3_{Tr}$ .... so on. till l_i and l_{i+1} are almost same (when 2 hyperplanes are very close to each other)

$D^{Tr}_{i+1} \rightarrow$ outlier free data
$l_{i+1} \rightarrow$ outlier robust model

Youtube: Programming Cradle

⊛ Retraining models periodically.

eg: predict stock prices for the next day.



productionized model (M_{10}) on Day 11

let

		AUC
Day 11 →	M_{10} →	0.8
Day 12 →	M_{10} →	0.81
Day 13 →	M_{10} →	0.6

huge drop in auc.

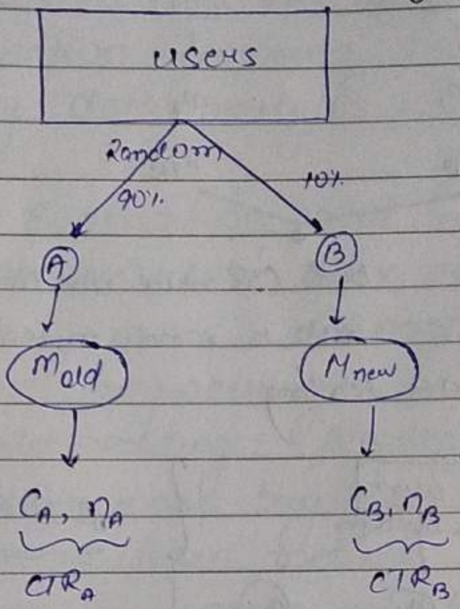
⊛ How to determine if we have to retrain?

→ model performance dropped
→ dataset / data pts. changed. (non stationary data)

if retraining cost is not too much retrain periodically and period can be small. as small as 1min, 1hr.

⊛ **A/B Testing** :- Bucket testing or Split run or Controlled exp.

⊛ Eg:- Develop model for Google predict to which ad to show for a search Query.



∴ C_A = no. of clicks
n_A = no. of times shown.

click through rate = $\frac{C_A}{n_A}$

now, check CTR_A and CTR_B for few days
if $CTR_A < CTR_B$
then we can say new model is better but this is not, this straight forward.

Youtube: Programming Cradle

→ Ⓐ

C_A	n_A
100	100K

CTR = $\frac{100}{100K}$ = 1%

Ⓑ

C_A	n_A
2	100

CTR = $\frac{2}{100}$ = 2%

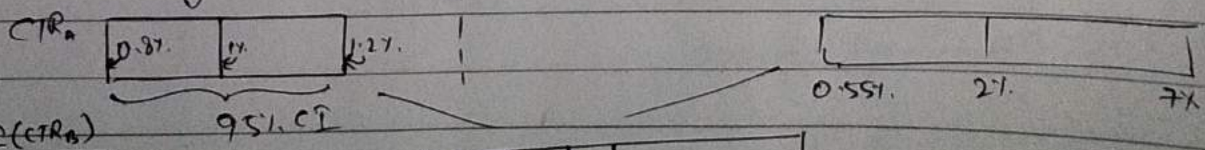
∴ even though CTR of B is greater than A we can not say model B or new model is good bcoz model B is over all showing less ads.
Hence we use confidence Interval.

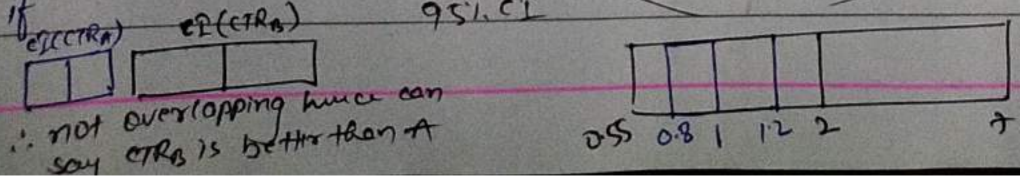
95% CI

CTR_A = $\underbrace{[0.8\%, 1.2\%]}_{\text{tighter}}$

95% CI

CTR_B : $\underbrace{[0.55\% ; 7\%]}_{\text{wide}}$

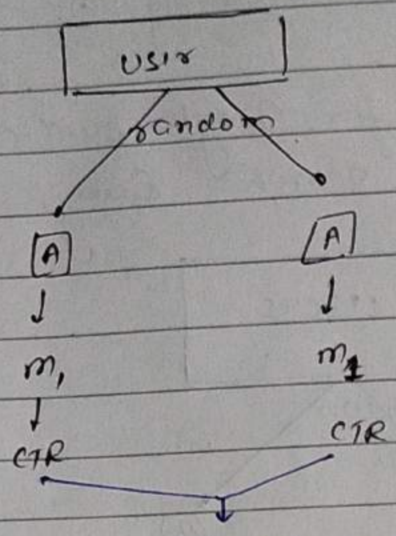




∴ not overlapping hence can say CTR_B is better than A

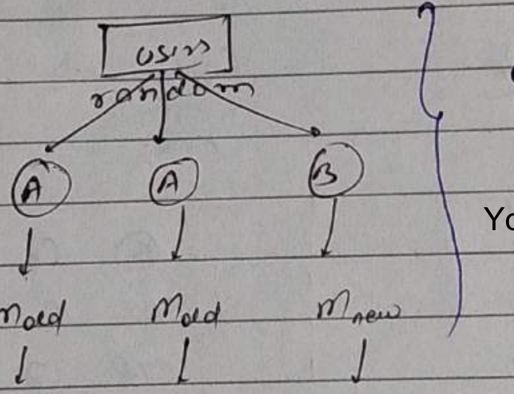
→ completely overlapping hence can't conclude

* AA testing
↓
tells the variance of same model on diff data



both CTR need not be same as data is randomized.

→ use of AA testing with AB testing



A-A-B test

Youtube: Programming Cradle

$CTR_A = 5\%$ $CTR_{A'} = 5.5\%$ $CTR_B = 5.4\%$

if we compare CTR_A and CTR_B we may conclude CTR_B is better but when we see all CTR_A, $CTR_{A'}$ and CTR_B we can conclude this improvement might be just bcoz of randomization

⊛ Data Science / ML project life-cycle

1.) understand business requirements :- define the problem, customer.
2.) Data aquisition: ETL, SQL, [DB, DW, log-files, Hadoop/spark]
3.) Data preperation: cleaning, pre-processing.
4.) Exploratory data analysis : plots, vizⁿ, hypothesis testing, slice and dice data
5) Modeling, Evaluation and Interpretation.
6) Communicate results: clear and simple to understand (1 page - 6 page descriptⁿ)
7) Deployment : engineering.
8) Real world testing : A/B testing.
9) Customer / business buy-in, convince them.
10) Operations :- retrain models, handle failures, process
11) Optimization : improve models, more data, more features, code optimization

⊛ VC-dimension :- Vapnik - Chervonentis

Q) How powerful is a $\underbrace{\text{class of models}}$?

→ Linear - models
→ boosting algo.
→ NNs

VC - dimension :- Statistical ML, computation learning theory.
It is a theoretical way to decide which model is better. (not much used in applied / practical)

VC - dim :- max no. of pts. that can be shattered/ separated by a model for all possible config.

VC - dim (lr model) = 3 → 3 is the max^m no. of pts that can be separated by a lr model for all possible config.

VC - dim (RBF - SVM) = ∞

⟹ if VC - dim (M_1) > VC - dim (M_2)

then

theoritically M_1 is powerful than M_2 but doesn't imply same in practical.