Topics

- Elastic Net -reg (L_1-reg)

Misc:

- Calibration : Platt scaling ; Isotonic Regr
- RANSAC : Robust models
- Lift & gain charts } - DA: Models
- Normal Equations for Linear Regression

GMM & EM:

- MLE , MAP & Bayesian estimation for Lr. Reg..
- Kmeans ++ (speedup)

@ a speed that you are comfortable

Recap: L_1-reg

hinge-loss; Huber-loss

log-loss; sq.loss

L_2-reg:

$$\min_{w_j} \; \text{loss} + \lambda \underbrace{\sum_{j=0}^{d} (w_j^2)}_{\|w\|_2^2}$$

reg: avoid overfitting

$$\boxed{w^T x_i + w_0}$$

L1-reg :-

$$\min_{w_j} \; \text{loss} + \lambda \underbrace{\overbrace{\sum_{j=0}^{d} |w_j|}}$$

$\|w\|_1$

L1-reg introduces sparsity in w ⇒ all the less useful fj's weights will become zero

1 d > n
almost n weights non zero

2 correlated features
only one w_j = non-zero
rest all zero

(intuition in prev. class)

[$(\lambda_1 \& \lambda_2)\uparrow$ ⇒ avoid overfitting]

Elastic Net:

$$\min_{w_j} \; \text{loss} + \left[\boxed{\lambda_1} \underbrace{\sum_{j=0}^{d} w_j^2}_{L_2} + \boxed{\lambda_2} \underbrace{\sum_{j=0}^{d} |w_j|}_{L_1} \right]$$

L_2 ↓ Ridge Tikhonov

L_1 → Lasso

[{ $\lambda_2 \uparrow$ as compared to λ_1

→ more sparsity]

λ_1 , $\lambda_2 \uparrow$: reduce overfitting

$$\min_{w_j} \; loss + \boxed{\lambda_1} \sum_{j=0}^{d} w_j^2 + \left[\boxed{\alpha} \cdot \lambda_1\right] \sum_{j=0}^{d} |w_j|$$

$\alpha \uparrow \Rightarrow \alpha\lambda_1 \uparrow \Rightarrow$ more sparsity

$\lambda \uparrow$: less overfit

$\alpha \uparrow$: " " + more sparse models

6 / 10

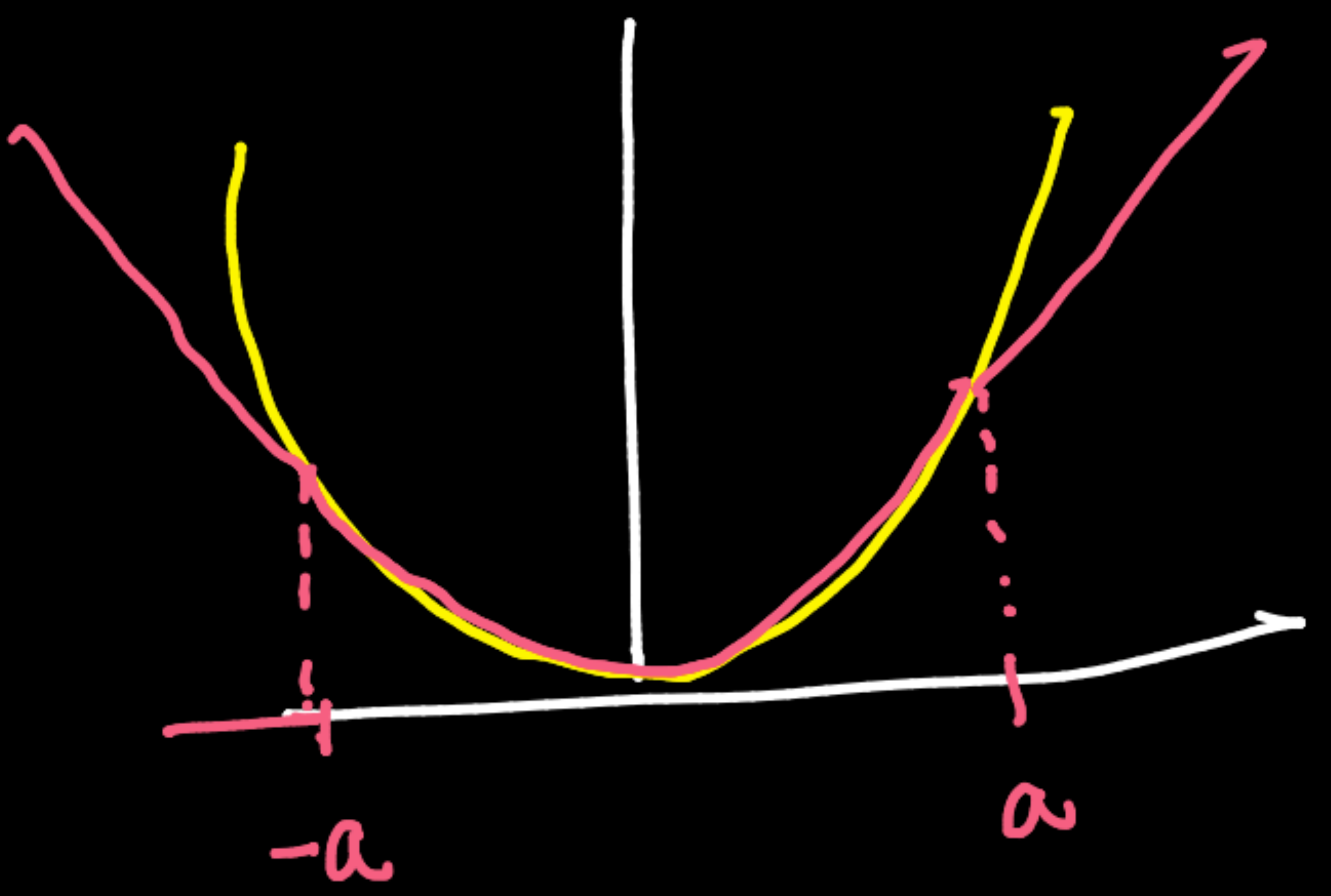
Q

Huber-loss:- L_1 & L_2 reg ✗

less impacted by outliers

modified sq. loss for robustness

$$\min_{w_j} \boxed{\text{Huber loss}} + \lambda_1 L_1 \text{reg} + \lambda_2 L_2 \text{reg}$$



$e_i = y_i - \hat{y}_i$

$$\min \; \text{loss} + \lambda(\text{reg})$$

log-loss

sq. loss

Huber.loss

Hinge-loss

$L_1 \mid L_2 \mid EN$

$\boxed{L_p\text{-norms}}$

harder to optimize over

scikit-learn 1.1.1
Other versions

Please cite us if you use the software.

sklearn.linear_model.LogisticRegression
Examples using **sklearn.linear_model.LogisticRegression**

supported by the 'saga' solver.

Read more in the User Guide.

Parameters:

penalty : *{'l1', 'l2', 'elasticnet', 'none'}, default='l2'*

Specify the norm of the penalty:

- ~~'none'~~: no penalty is added;
- 'l2': add a L2 penalty term and it is the default choice;
- 'l1': add a L1 penalty term;
- 'elasticnet': both L1 and L2 penalty terms are added.

> **Warning:** Some penalties may not work with some solvers. See the parameter `solver` below, to know the compatibility between the penalty and solver.

New in version 0.19: l1 penalty with SAGA solver (allowing 'multinomial' + L1)

dual : *bool, default=False*

Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Prefer dual=False when n_samples > n_features.

tol : *float, default=1e-4*

Tolerance for stopping criteria.

...pport vector machines, smaller values specify

Chrome File Edit View History Bookmarks Profiles Tab Window Help

Calibration

(practical)

esp with

→ imbalanced - data

logistic reg. model

or any binary classfn. model

$P(y_i = 1 \mid x_i)$

Task: are these probabilities **sensible**

11 / 12

✓ Calibration plot:

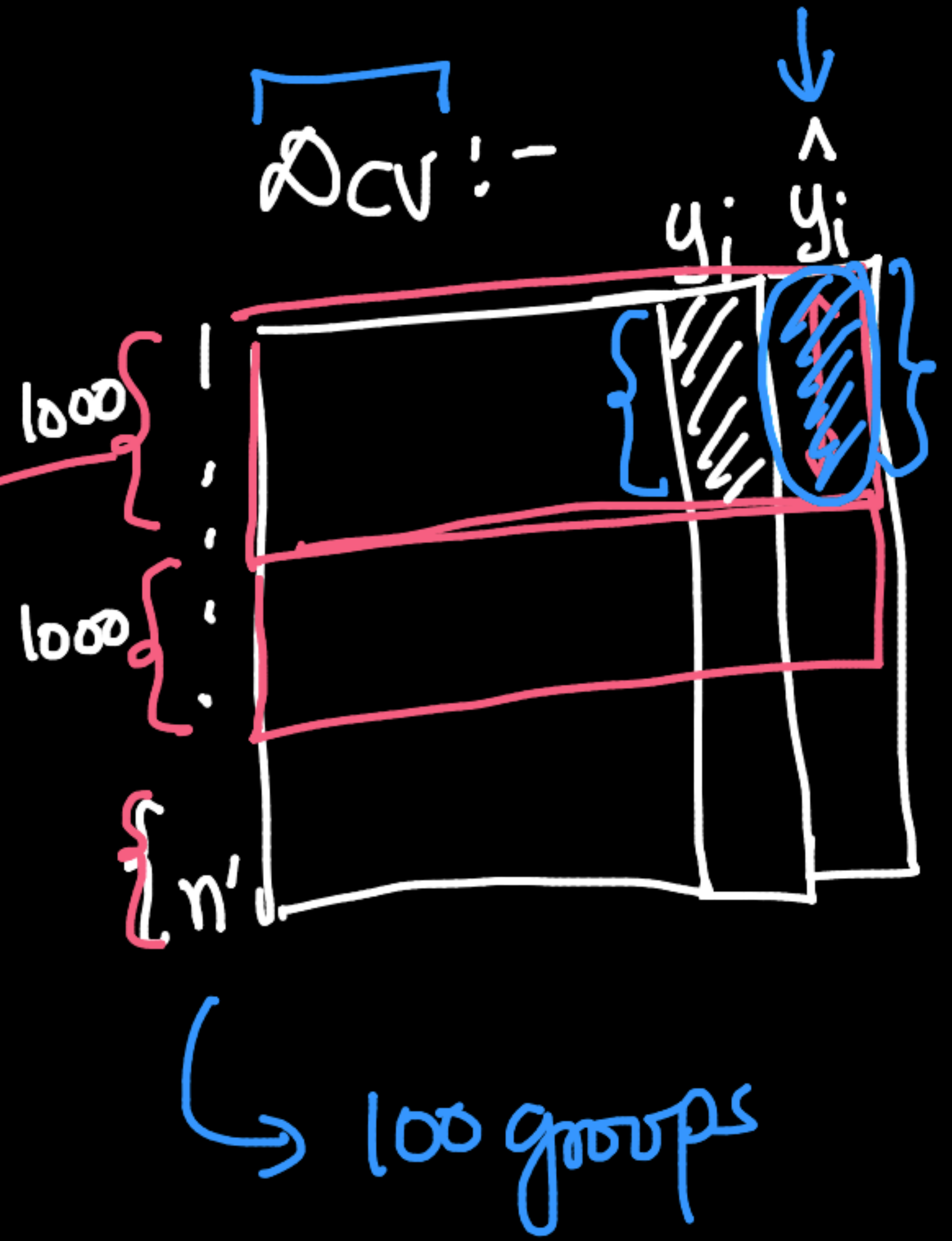
① Sort by $\hat{y}_i$ in incr. order

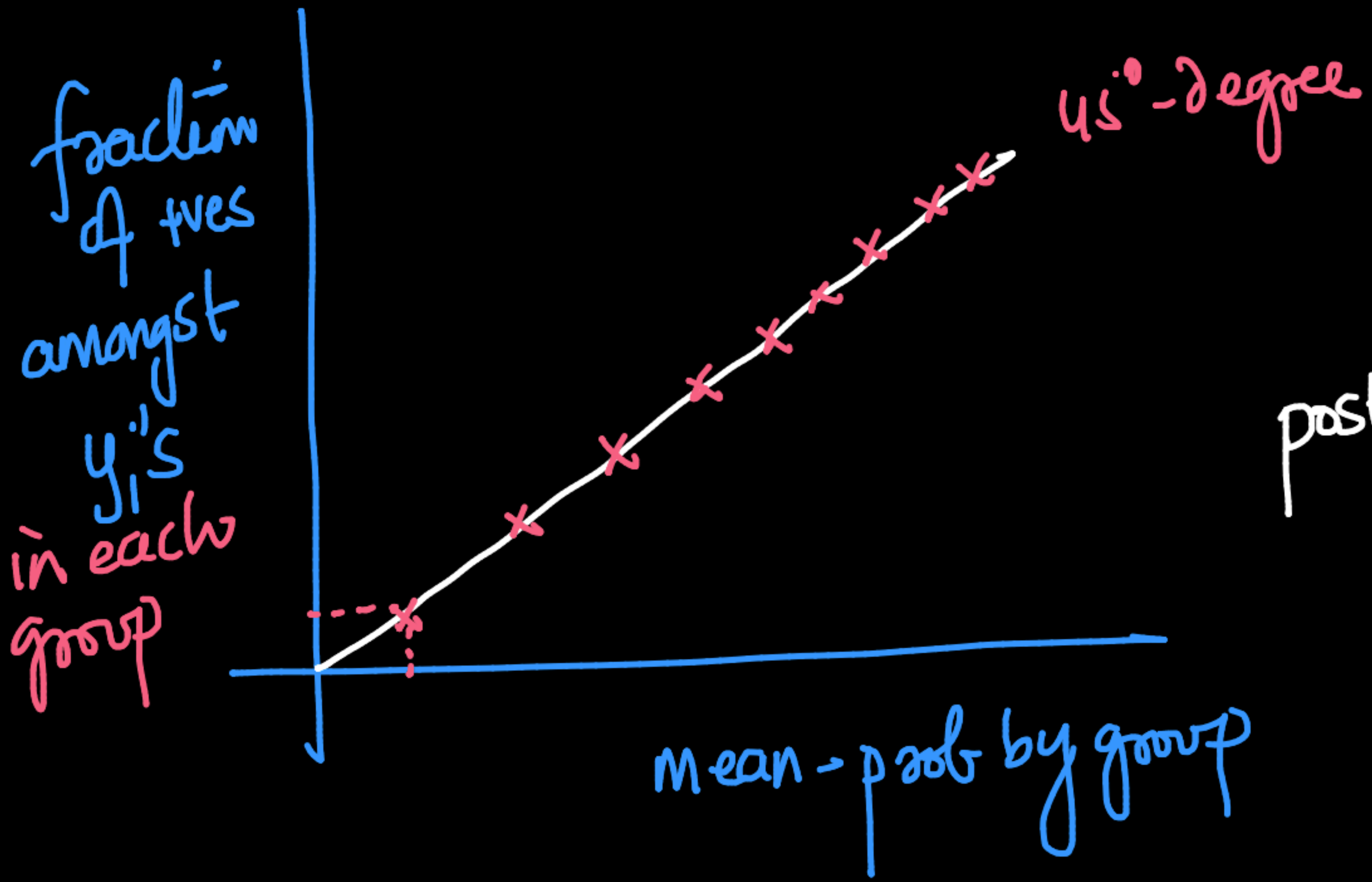
② $\forall$ group of sorted points:

↳ fraction of +ve in y_is → f_i

↳ mean of $\hat{y}_i$'s

↳



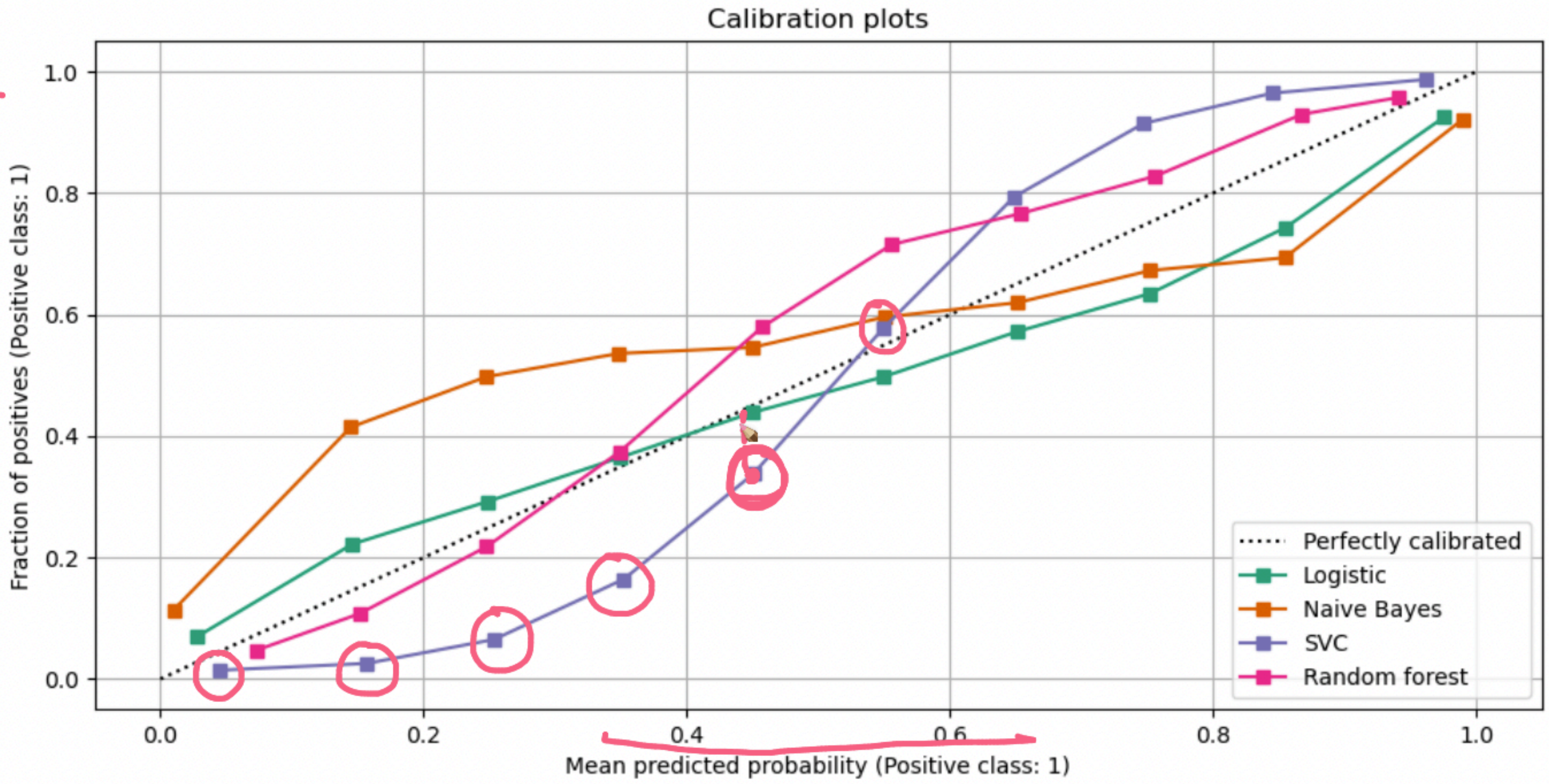


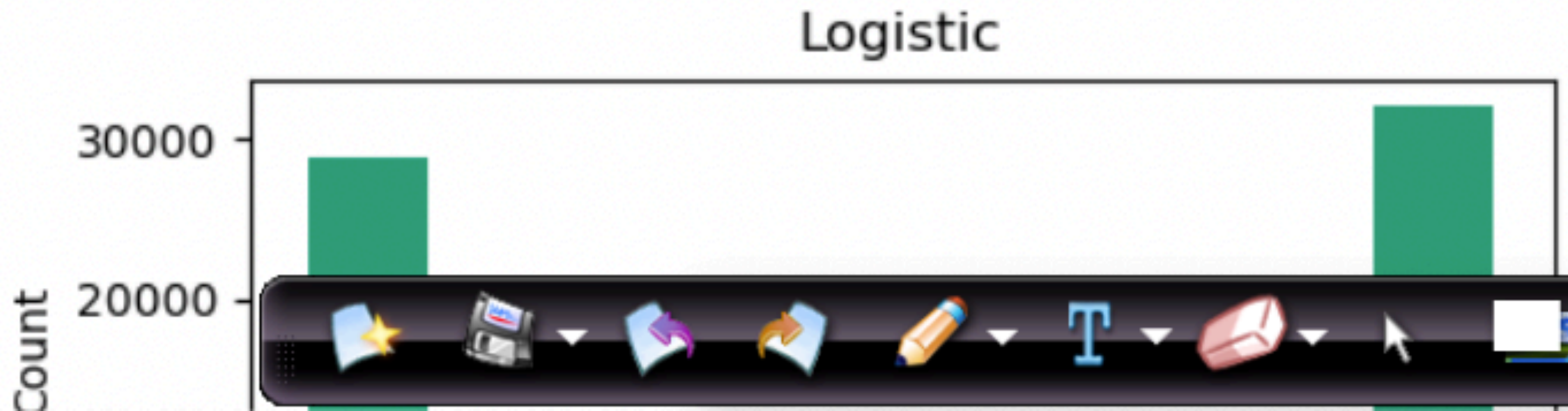
must-do
post-model-analysis

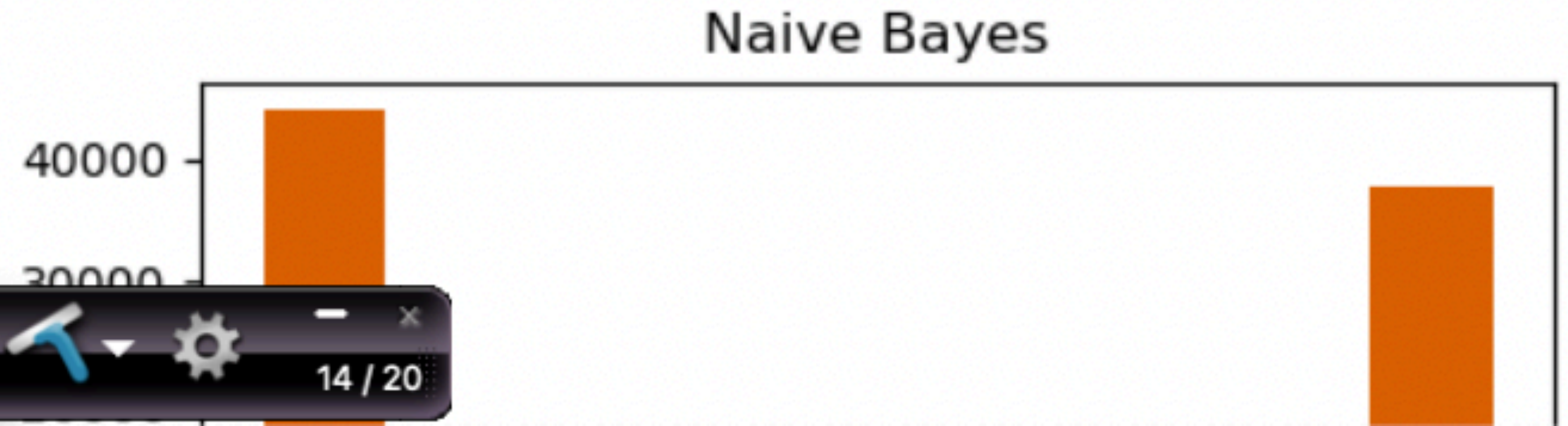
Prev Up Next

scikit-learn 1.1.1
Other versions

Please **cite us** if you use the software.





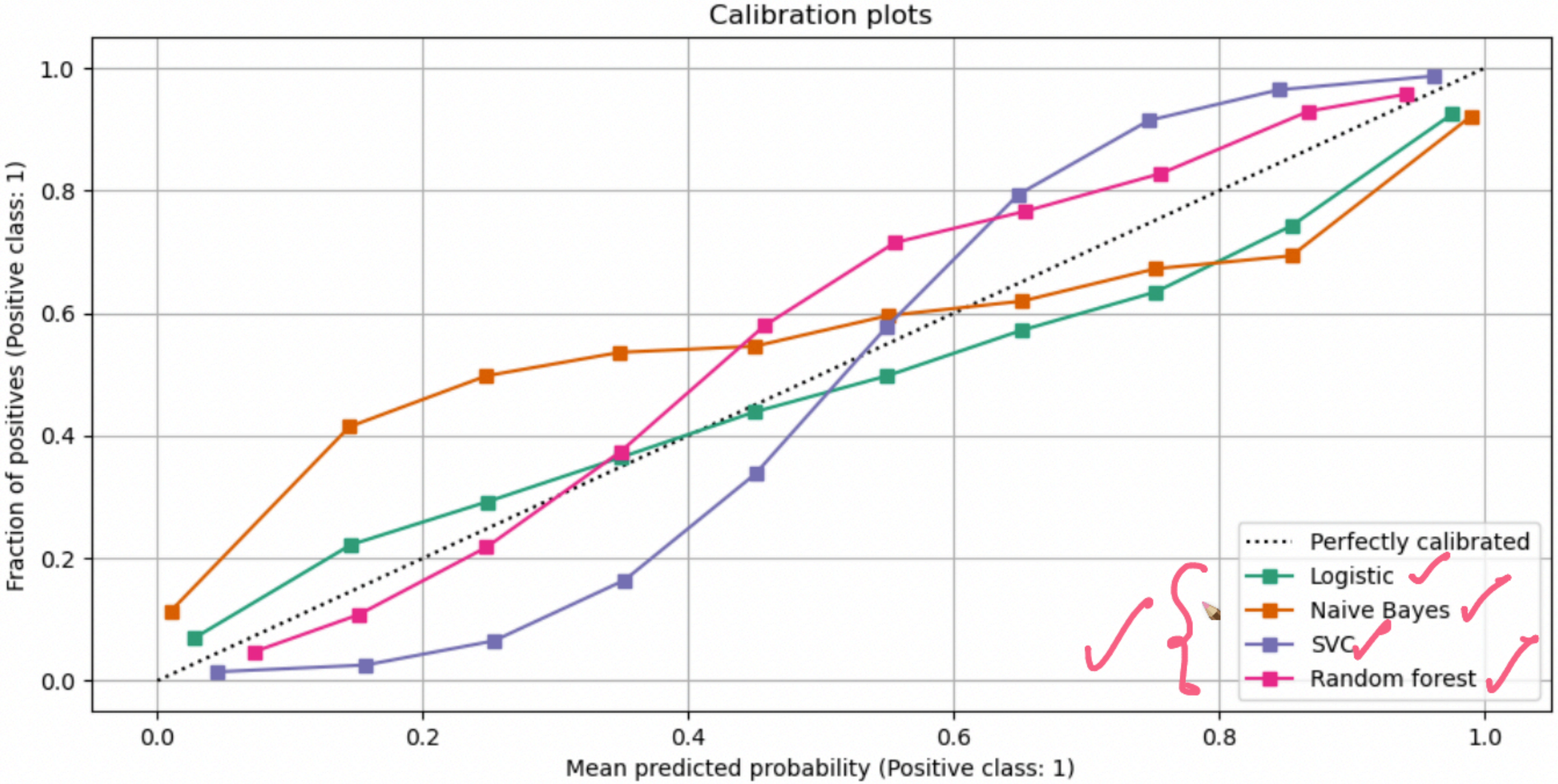


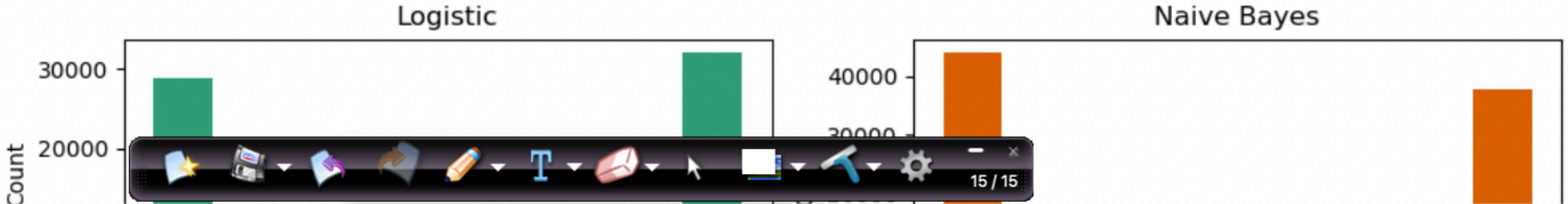
Toggle Menu

14 / 20

scikit-learn 1.1.1
Other versions

Please **cite us** if you use the software.





Toggle Menu

15 / 15

Safari File Edit View History Bookmarks Develop Window Help 10:02

Q

How do we fix this problem?

Platt's scaling (less-popular)

calib plot is S-shaped

Isotonic regression (Most popular)

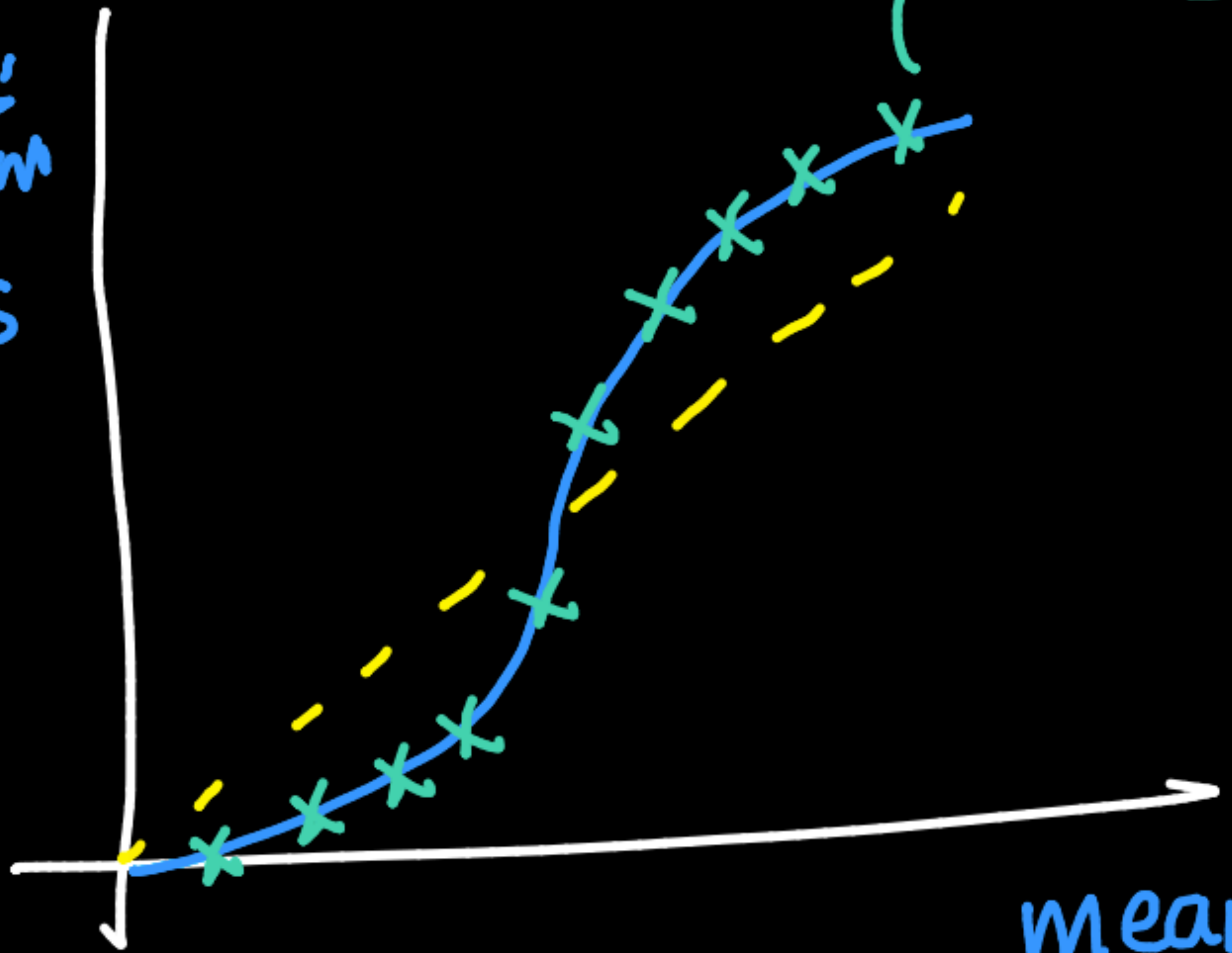
general purpose

16 / 22

Platt-scaling:

one group

fraction of 1's



Regression

mean-prob of $y=1$

Task:- $\hat{y}_i = \underbrace{P(y_i=1|x_i)}$ $\xrightarrow{\text{func}}$ $\underbrace{\text{corrected probability}}$

Another regression model → Calibration Model

Simpler

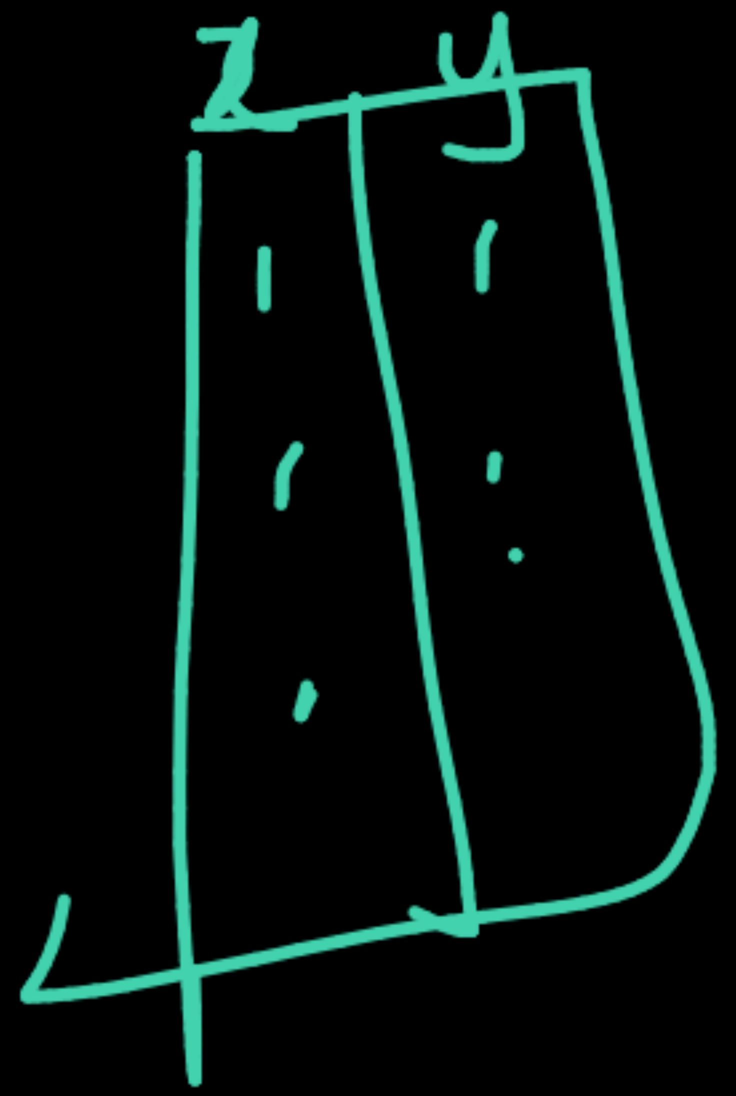
x_i → Class-model → $\hat{y}_i$ → M_{calib} → corrected probability

↓

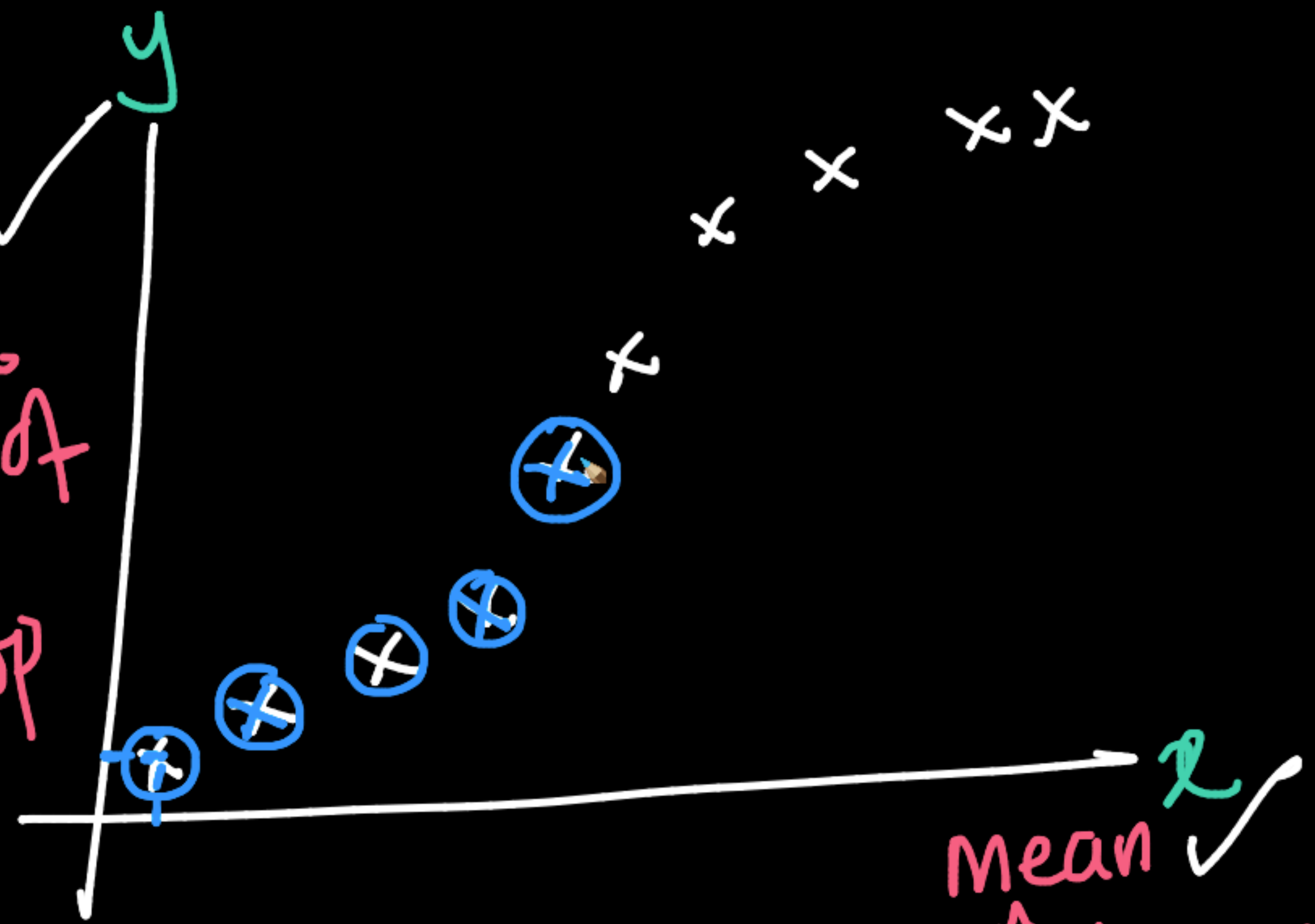
calib plots for this (45-degree line)

↓

full control any-model (build RF here)



actual fraction of +ves per group



mean $\hat{y}_i$'s per group

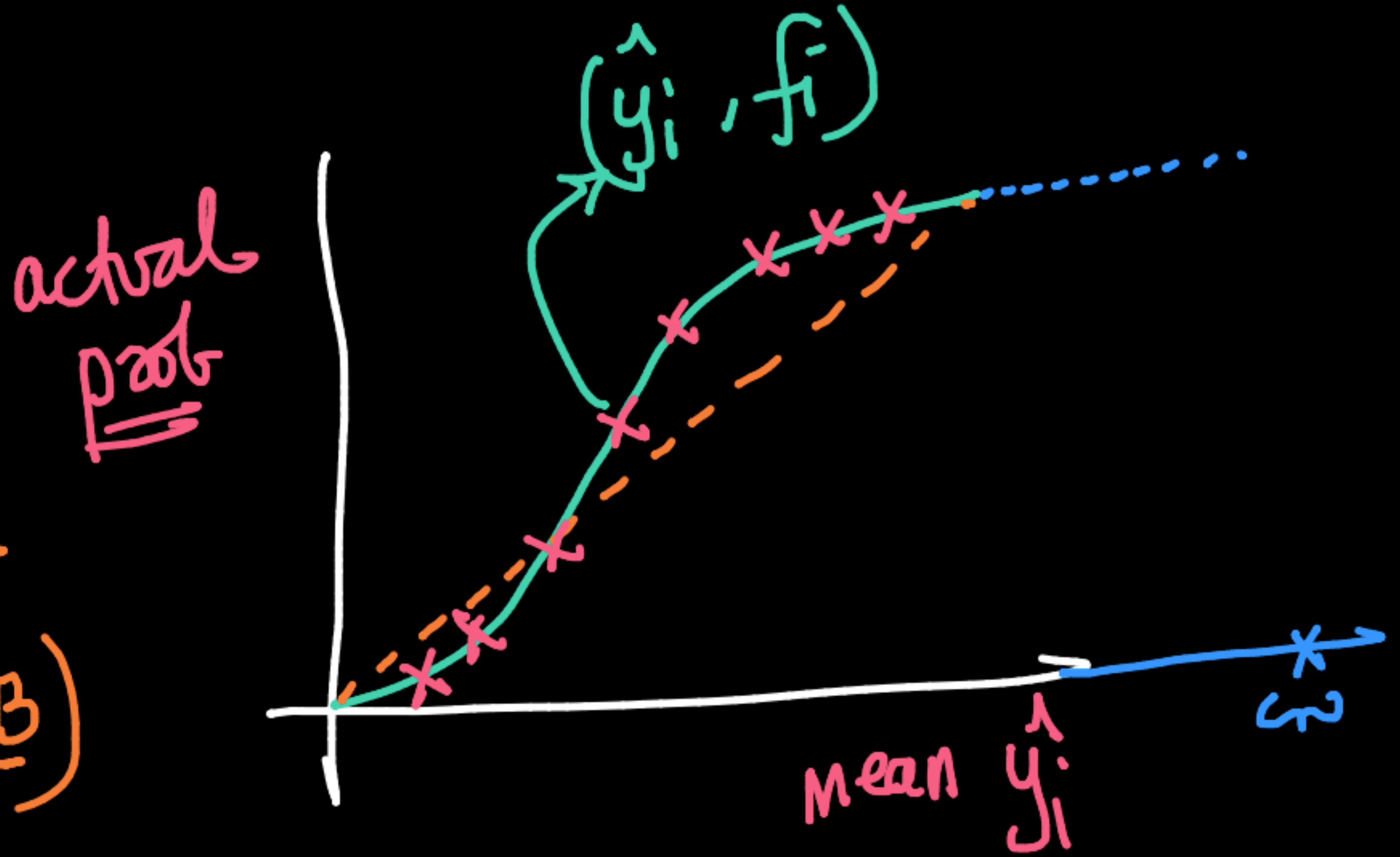
reduce the #gps

each gp has more pts

Mcalib

Sigmoid/Platt scaling:

$$f_i = \hat{P}(y_i = 1 | x_i) = \frac{1}{1 + \exp(A\hat{y}_i + B)}$$

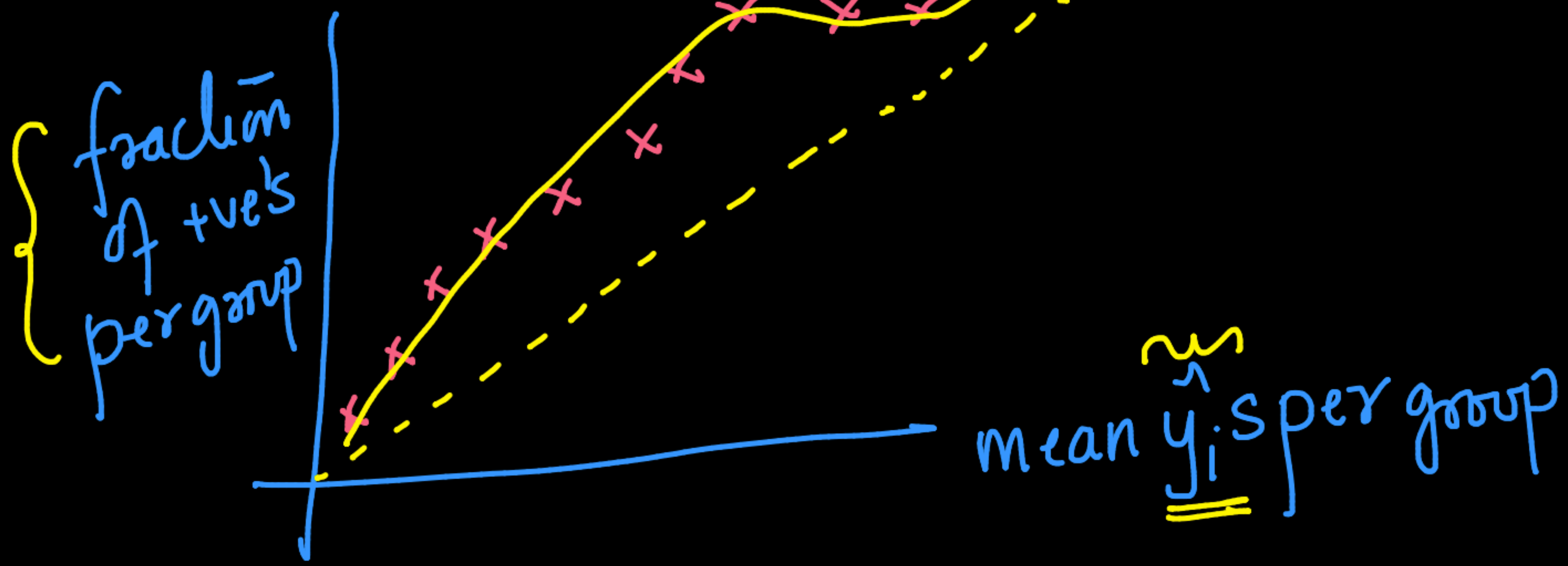


A & B ! $\rightarrow$ MLE/optimization

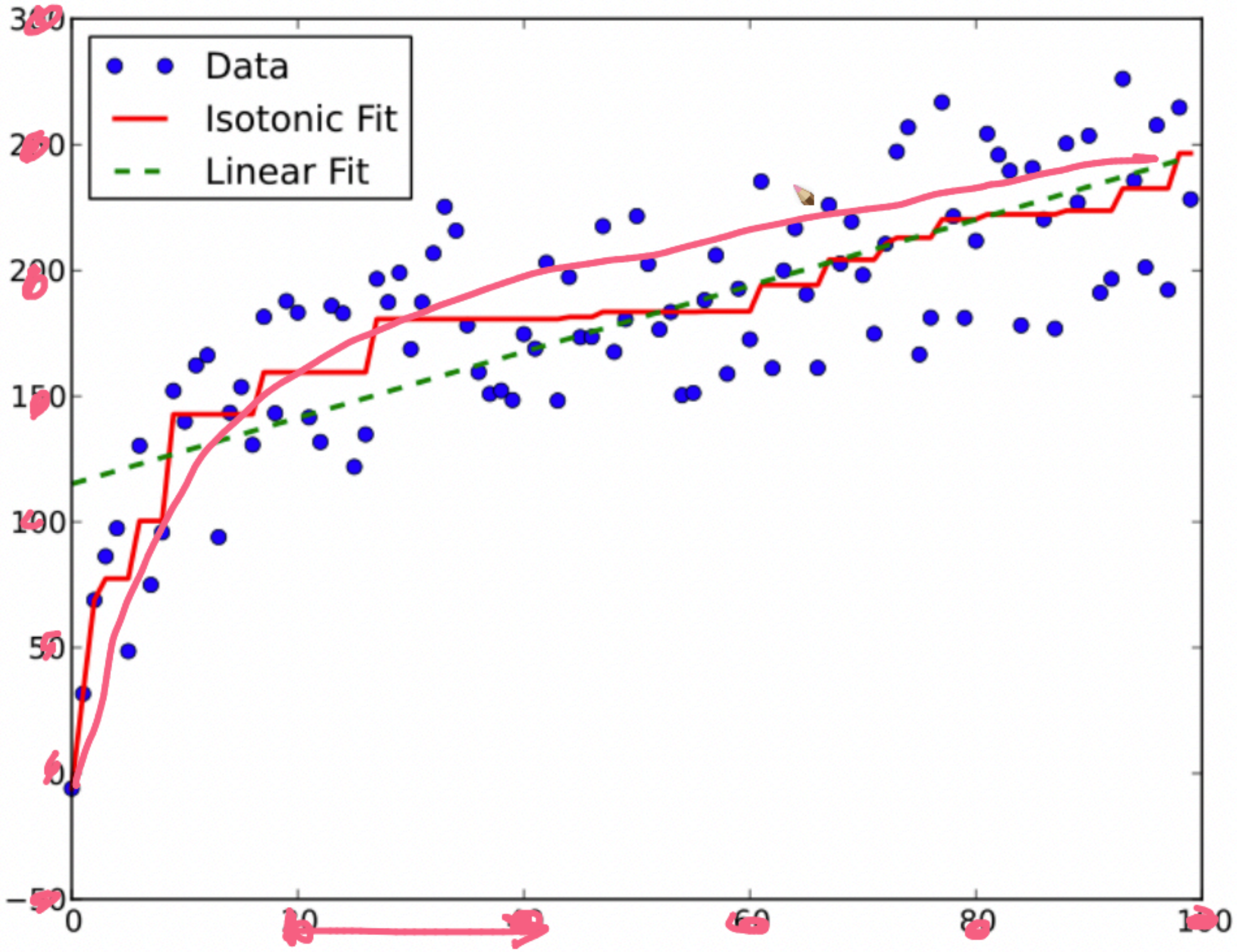
$$\min_{A,B} \left[f_i - \frac{1}{1+\exp(\underline{A}\hat{y}_i + \underline{B})} \right]^2 \quad : \underline{\underline{\text{SGD}}}$$

$$\hat{y}_i \longrightarrow f_i$$

Isotonic Regr:



en.wikipedia.org/wiki/Isotonic_regression#/media/File:Isotonic_regression.svg



f_i

mean $\hat{y}_i$'s

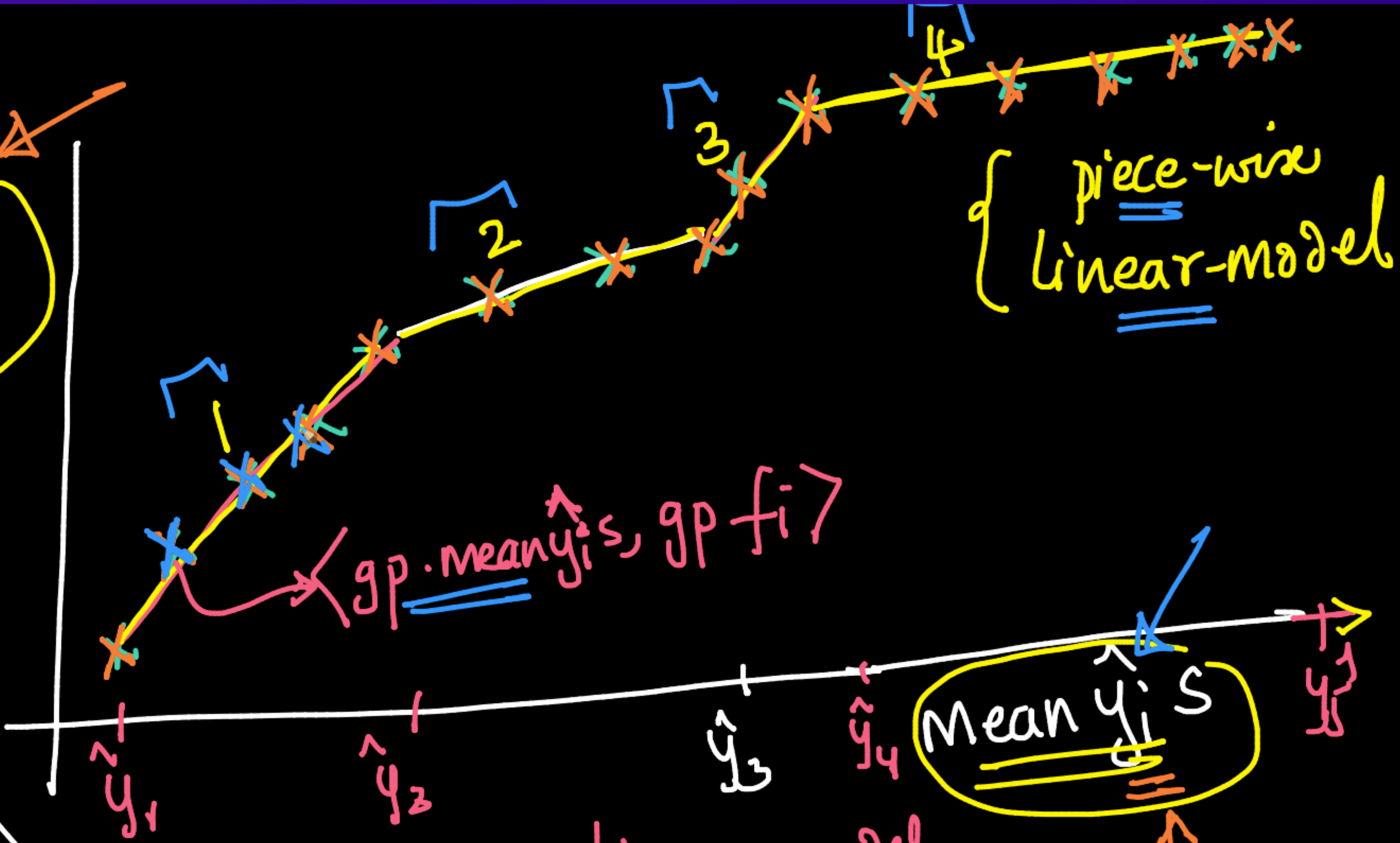
An example of isotonic regression (solid red line) compared to linear regression on the same data, both fit to minimize the mean squared error. The free-form property of isotonic regression means the line can be steeper where the data are steeper; the isotonicity constraint means the ...

More details

Alexeicolin - Own work

23 / 23

↑ f_i



to avoid overfitting

→ few groups

→ control / regularize the # of piece-wise models

piece-wise linear-model

Isotonic reg for calib

piecewise linear Model

$\hat{y}_i$

f_i

Safari File Edit View History Bookmarks Develop Window Help 10:25

$\left(\hat{y}_i, f_i\right)_{i=1}^{k=\#gps}$

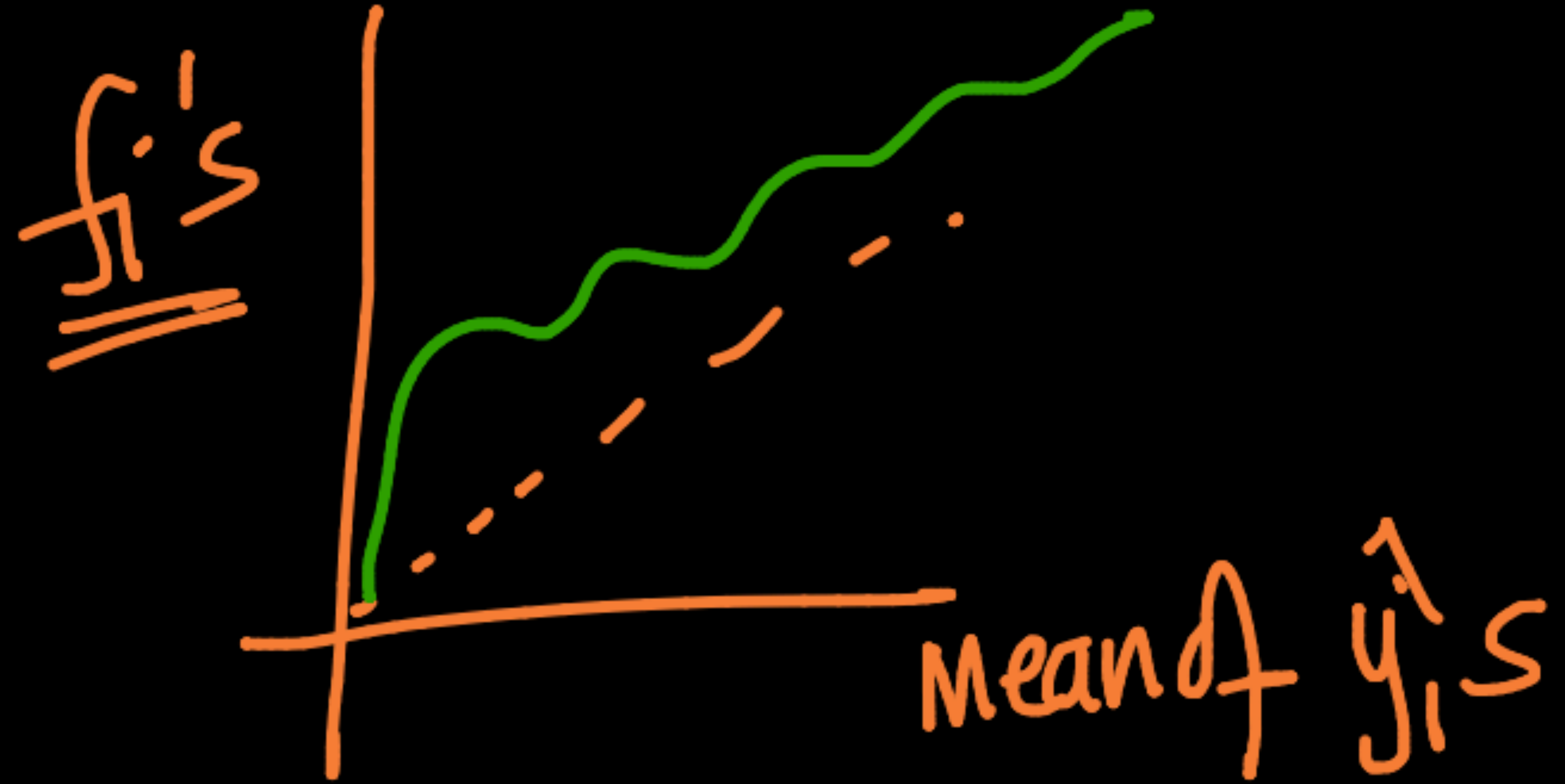
D Train

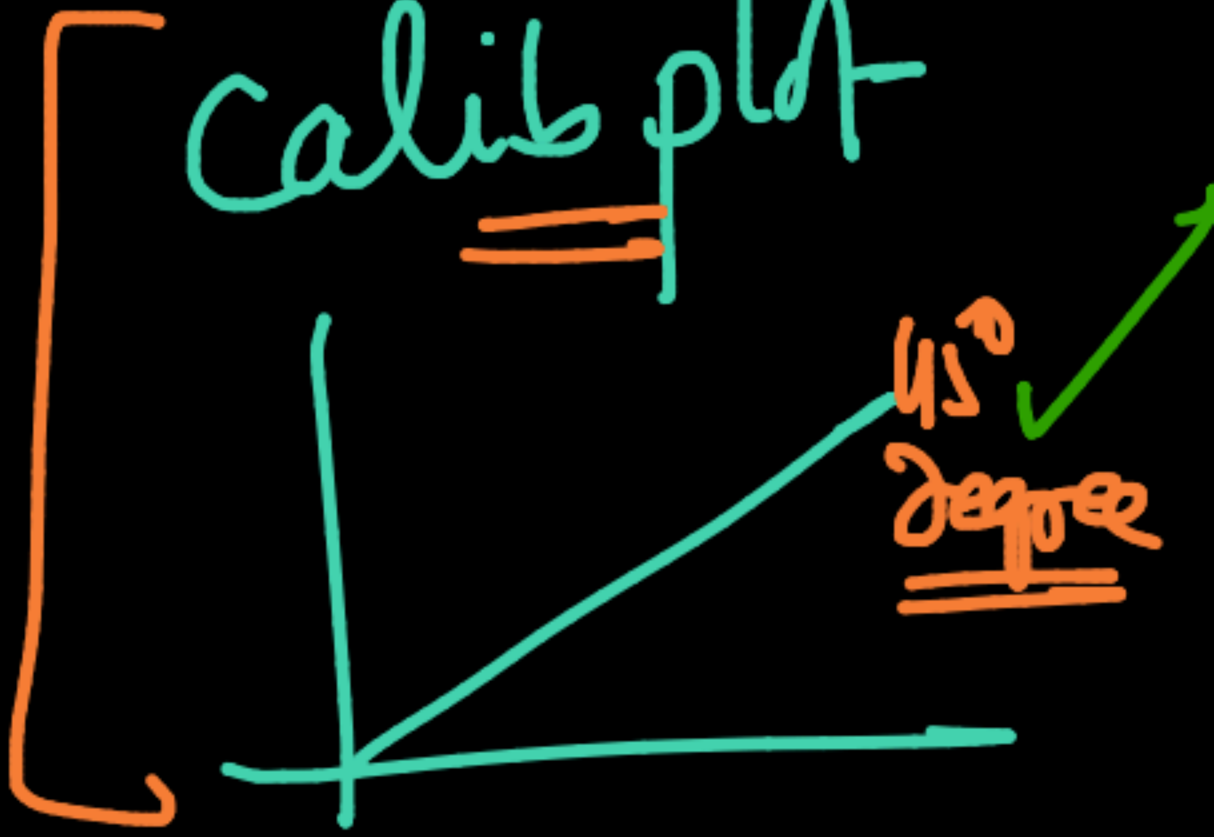
x_i → classfn.model → $\hat{y}_i$ → $\underbrace{M_{calib}}$ → $\boxed{f_i}$ → return

Isotonic

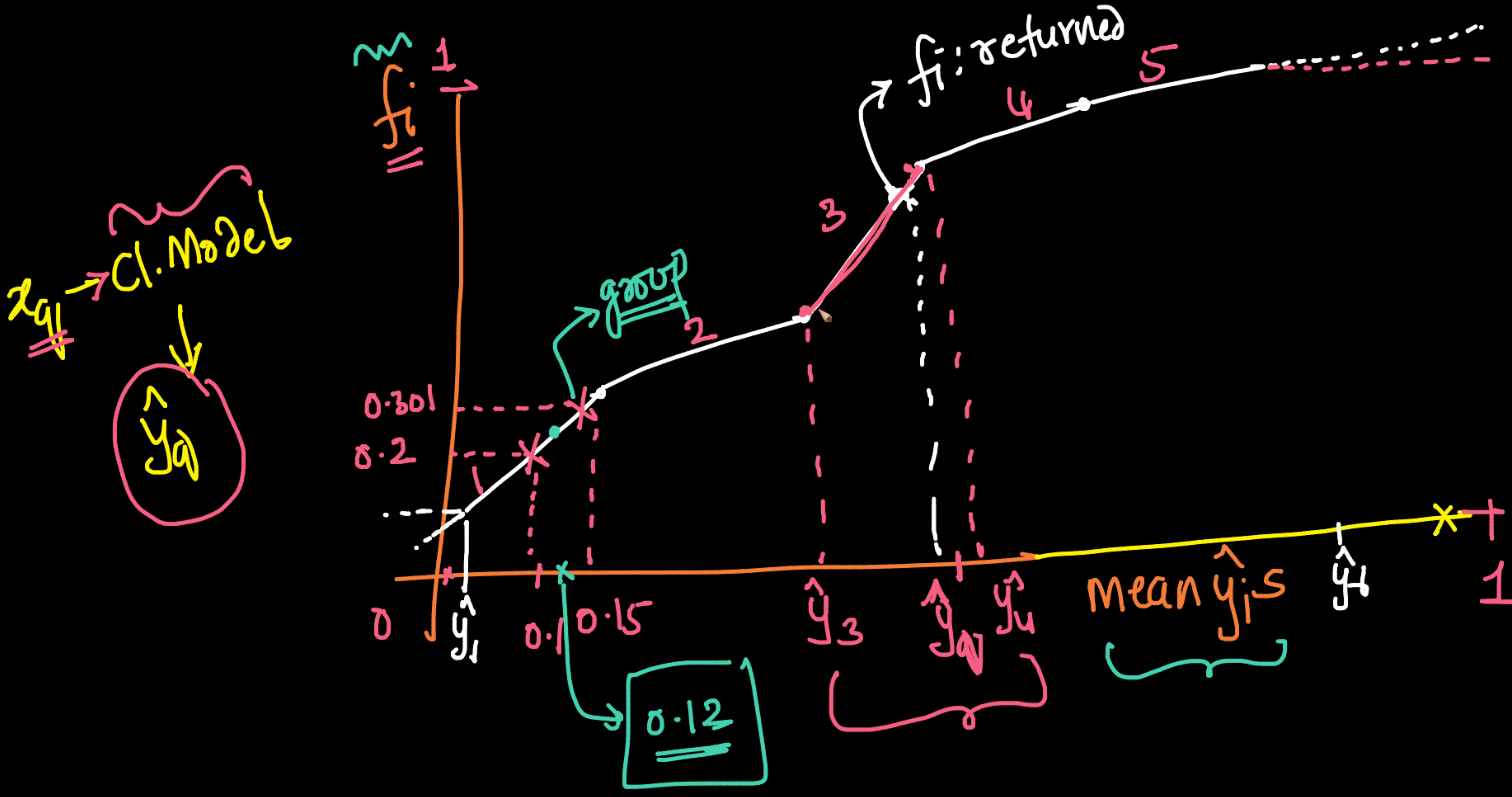
corrected prob

gp of Dcvs

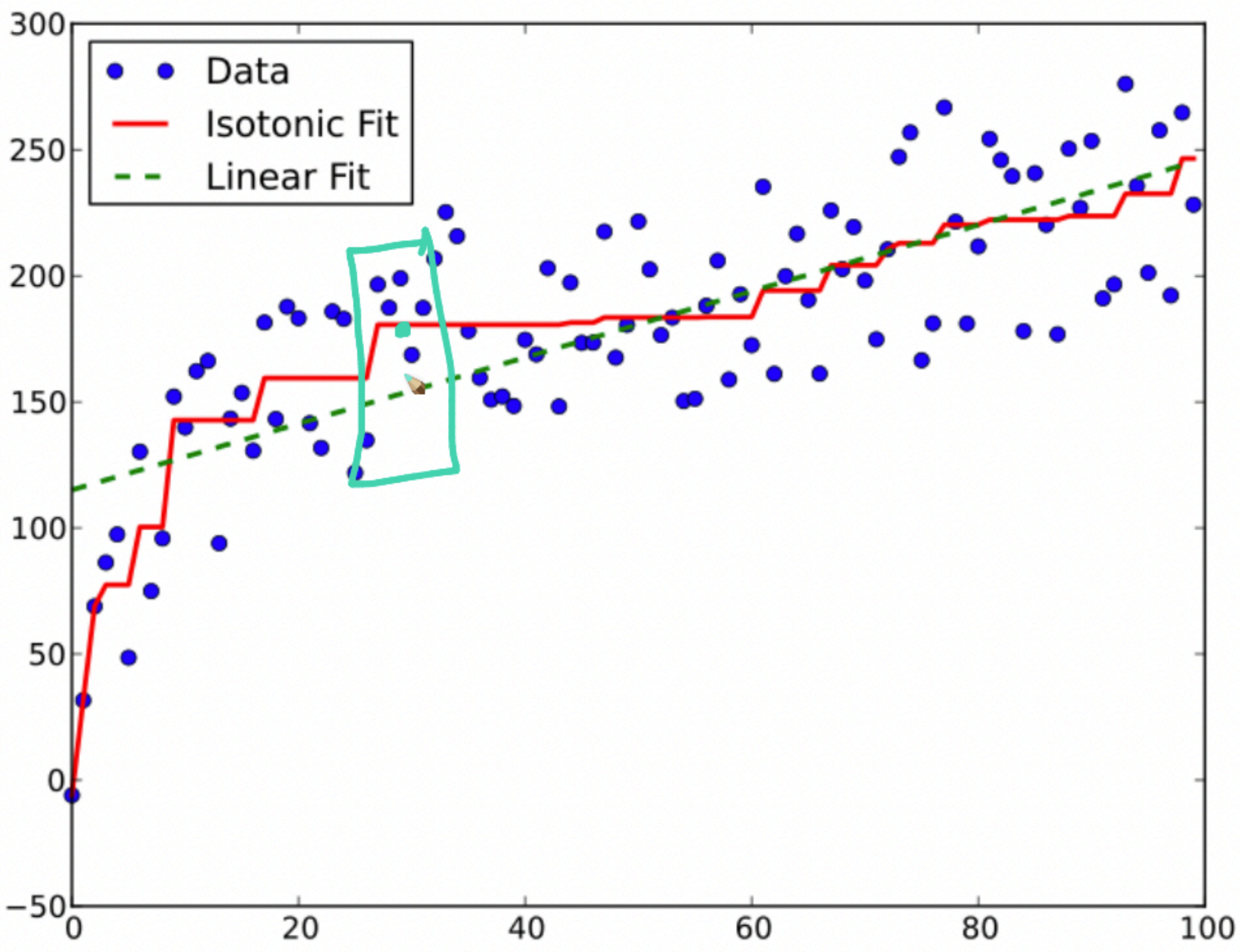




26 / 27



$x_q \rightarrow$ Cl. Model
$\hat{y}_q$
fi
1
fi : returned
1
2
3
4
5
group
0.301
0.2
0
$\hat{y}_1$
0.1
0.15
0.12
$\hat{y}_3$
$\hat{y}_q$
$\hat{y}_4$
mean $\hat{y}_i$s
$\hat{y}_H$
1



An example of isotonic regression (solid red line) compared to linear regression on the same data, both fit to minimize the mean squared error. The free-form property of isotonic regression means the line can be steeper where the data are steeper; the isotonicity constraint means the ...

More details

Alexeicolin - Own work

$$x_q \longrightarrow \overbrace{\text{Classfn. model}}^{\hat{y}_q} \longrightarrow \hat{y}_q \longrightarrow \overbrace{M_{calib}}^{\text{correct the prob.}} \longrightarrow f_q$$

$$\downarrow$$

return

References

Isotonic Median Regression: A Linear Programming Approach Nilotpal Chakravarti Mathematics of Operations Research Vol. 14, No. 2 (May, 1989), pp. 303-308

Isotone Optimization in R : Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods de Leeuw, Hornik, Mair Journal of Statistical Software 2009

Correctness of Kruskal's algorithms for monotone regression with ties de Leeuw, Psychometrica, 1977

Examples

```
>>> from sklearn.datasets import make_regression
>>> from sklearn.isotonic import IsotonicRegression
>>> X, y = make_regression(n_samples=10, n_features=1, random_state=41)
>>> iso_reg = IsotonicRegression().fit(X, y)
>>> iso_reg.predict([.1, .2])
array([1.8628..., 3.7256...])
```

gp. mean y's

fi

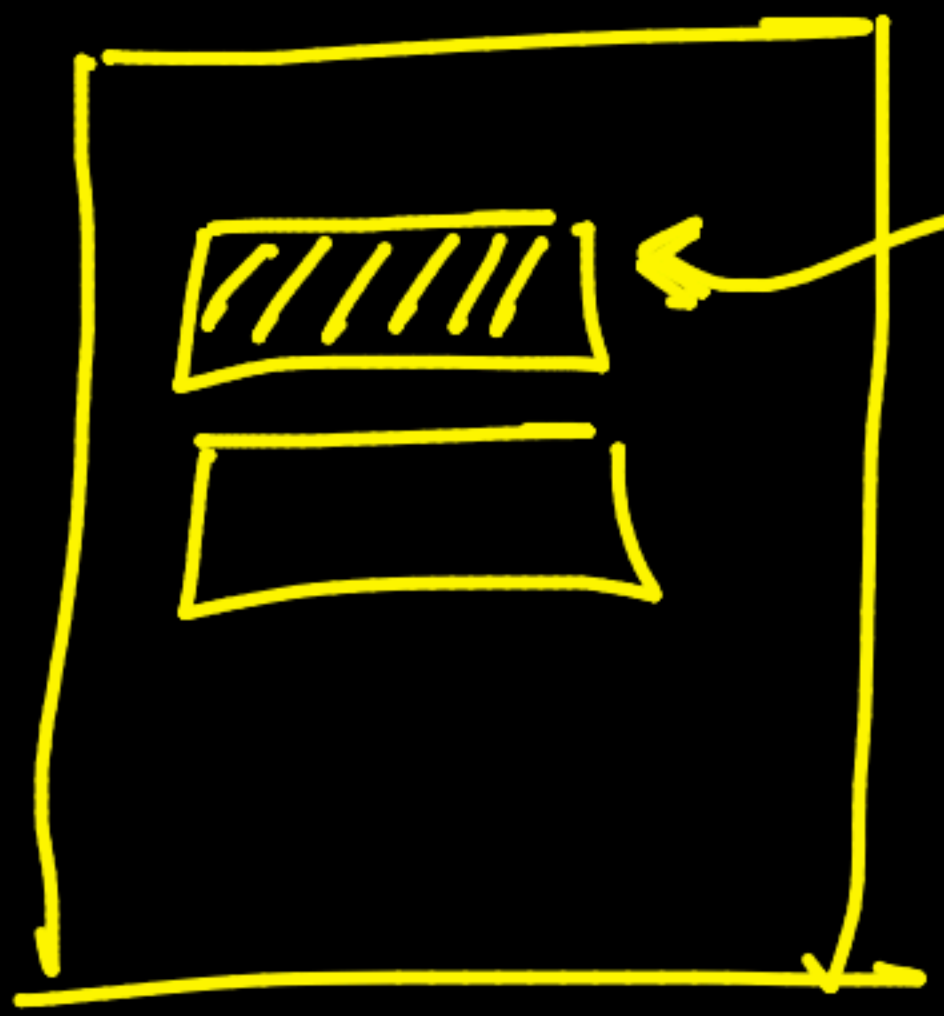
Methods

`fit`(X, y[, sample_weight])	Fit the model using X, y as training data.
`fit_transform`(X[, y])	Fit to data, then transform it.
`get_feature_names_out`([input_features])	Get output feature names for transformation.
`get_params`([deep])	Get parameters for this estimator.
`predict`(T)	Predict new data by linear interpolation.
`score`(X, y[, sample_weight])	Return the coefficient of determination of the prediction.
`set_params`(**params)	Set the parameters of this estimator.
`transfor`	

Toggle Menu

Correcting prob: f_i = more accurate

$$P(y_i = 1 \mid x_i)$$



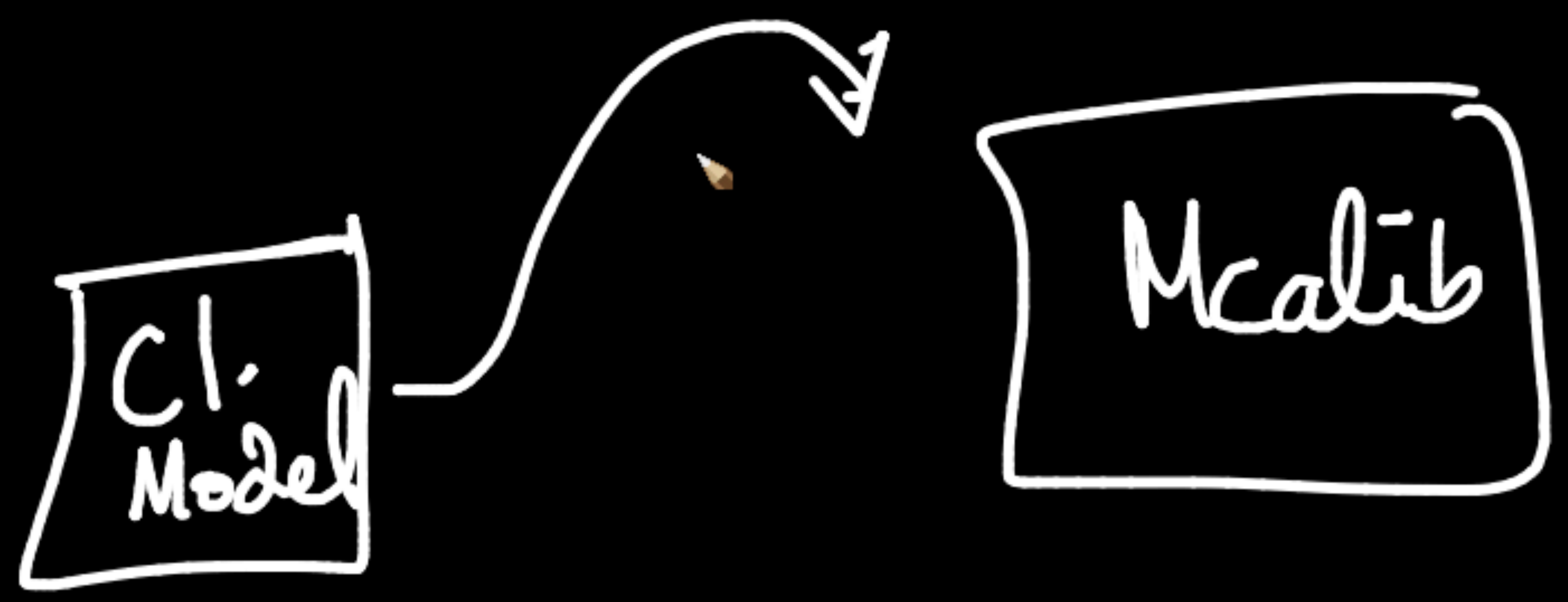
Click : 1

price paid by the adv

healthcare:-

next steps

$$P(y_i = 1 \mid x_i) = \underline{\underline{0.7}}$$


Cl.
Model
Mcalib

10:40

Robust: less prone to outliers

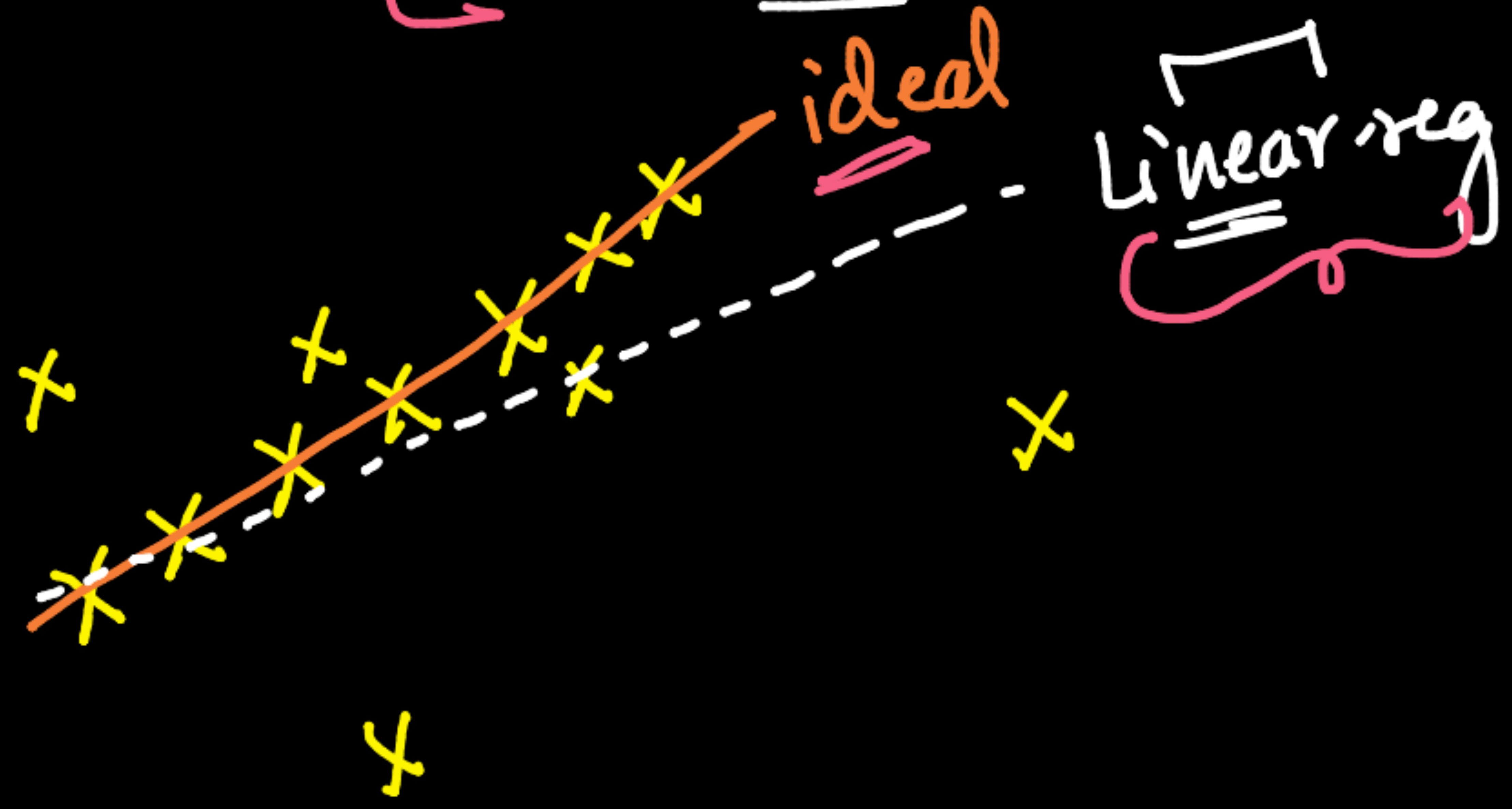
↳ Huber-loss in regr

RANSAC

Random Sampling Consensus.

[RANSAC: model agnostic

↳ general purpose



d-dim

100

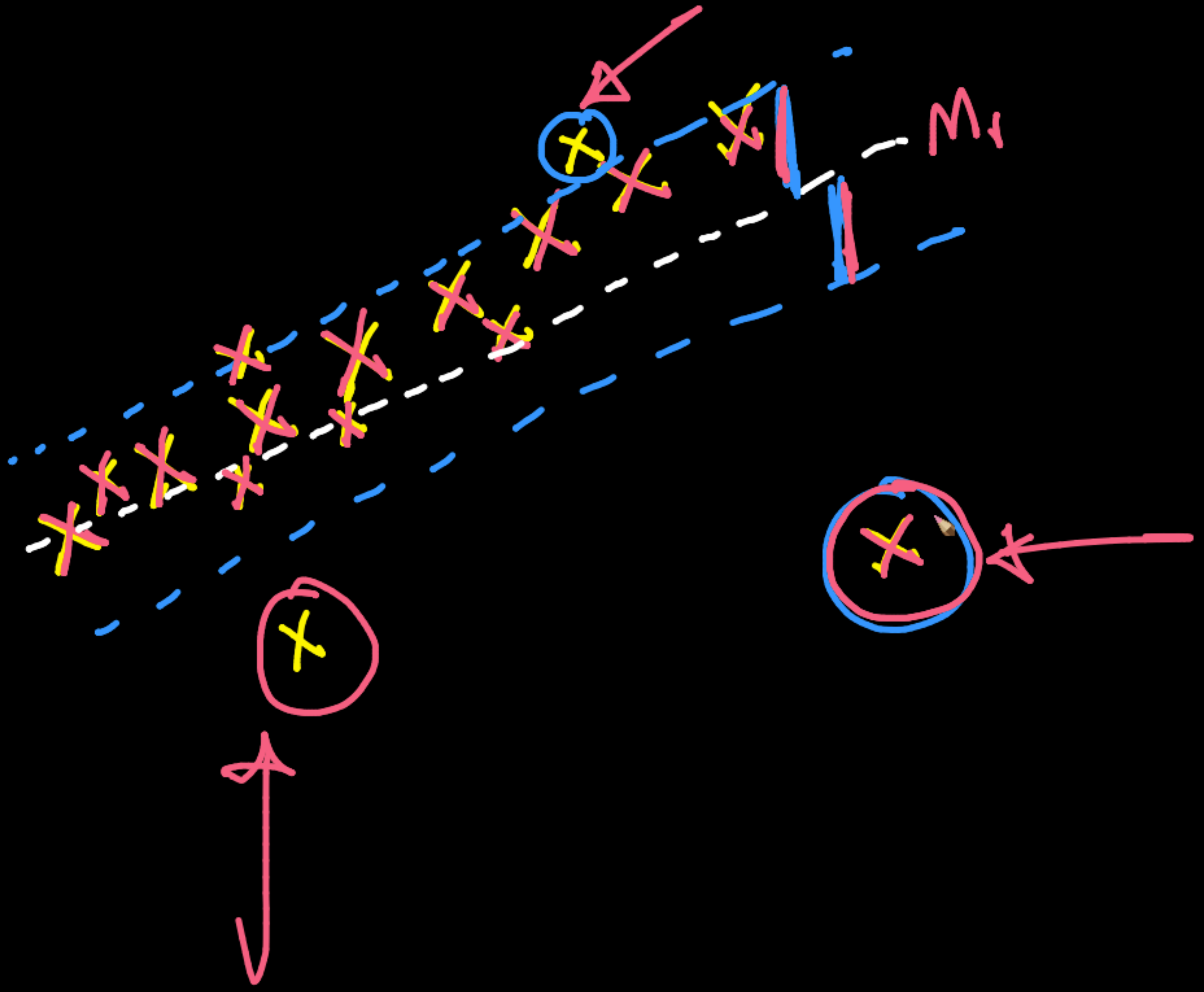
Input :- $\mathcal{D}_n = \{(x_i, y_i)_{i=1}^{n}\}$

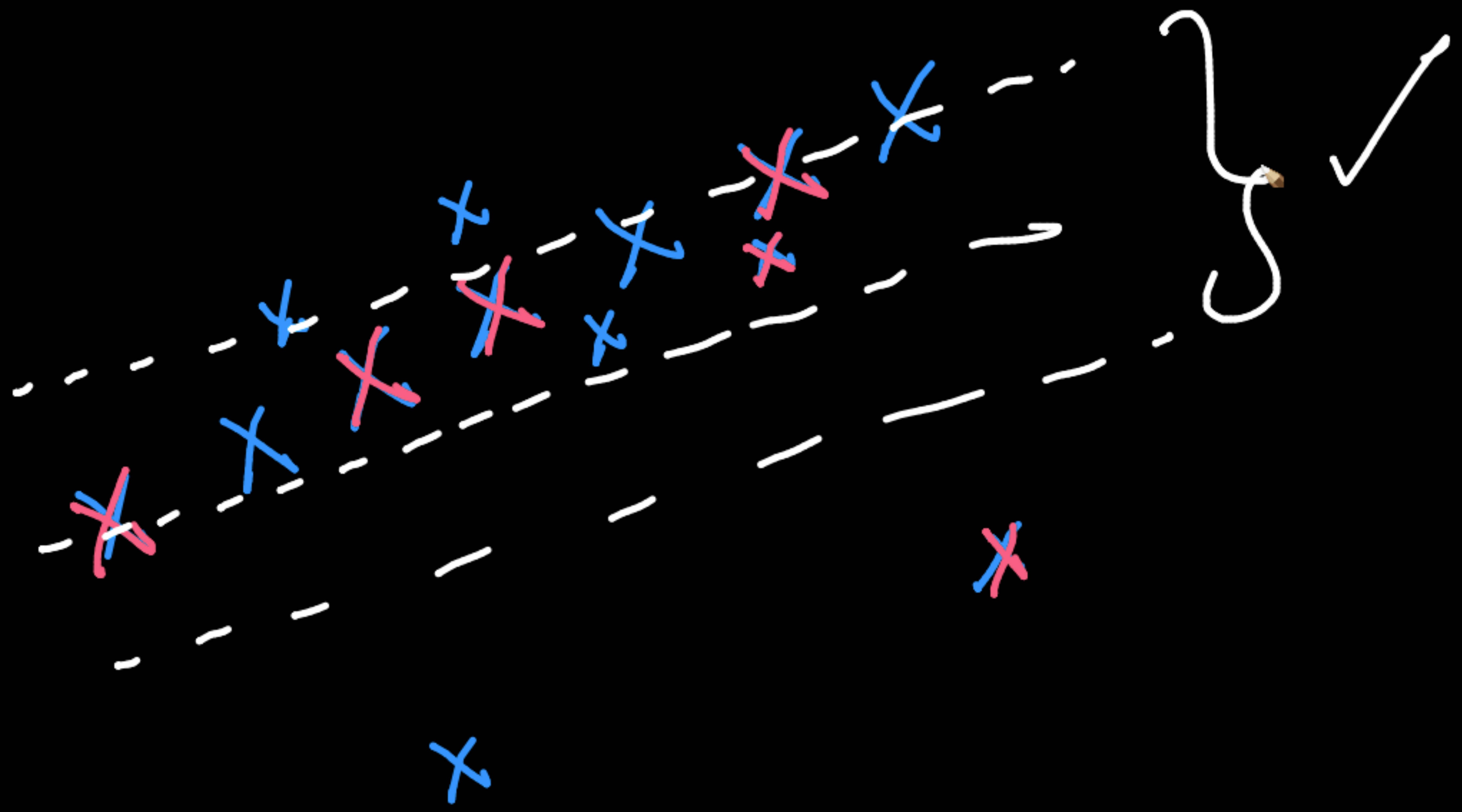
① select a random subset $\mathcal{D}'_{n'}$ from $\mathcal{D}_n$ — hypothetical inliers

$n' < n$

② fit my model (lr. reg) on $\mathcal{D}'_{n'}$

③ For all other points — $\mathcal{D}$ — alt: $\mathcal{D}-\mathcal{D}'$; is the error/loss on these pts small $\Rightarrow$ Consensus set





70% ?

In step(4) $\frac{|C|}{|D|} < th = 90\%$

↓

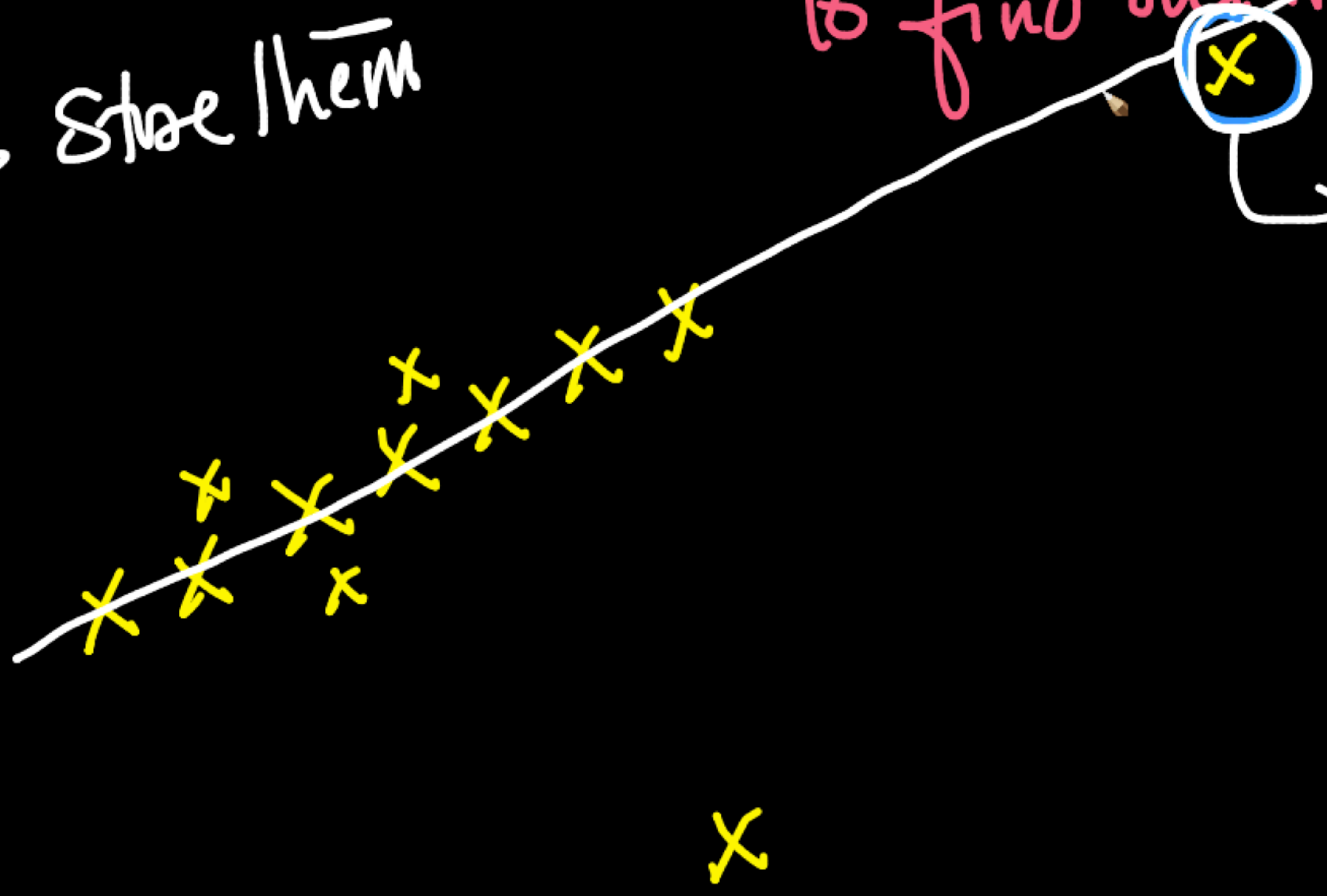
repeat from step 1 again

Safari File Edit View History Bookmarks Develop Window Help 11:00

41 / 41

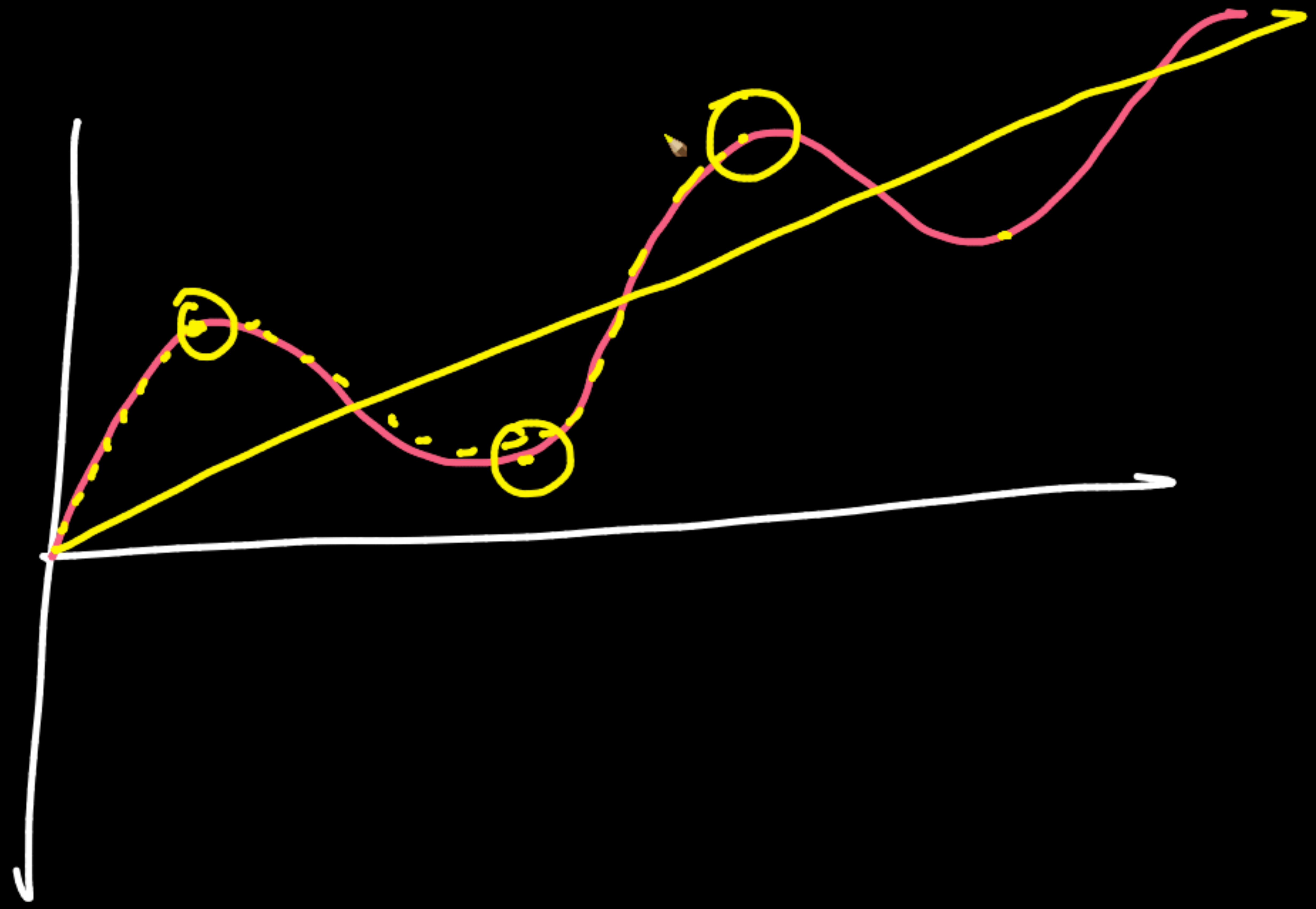
RANSAC → using the model & appropriate loss to find outliers

- Lr. reg → } → store them
- RF-reg
- ⋮



not a model based outlier

Safari File Edit View History Bookmarks Develop Window Help 11:04



42 / 42

RANSAC from scratch

↳ Homework

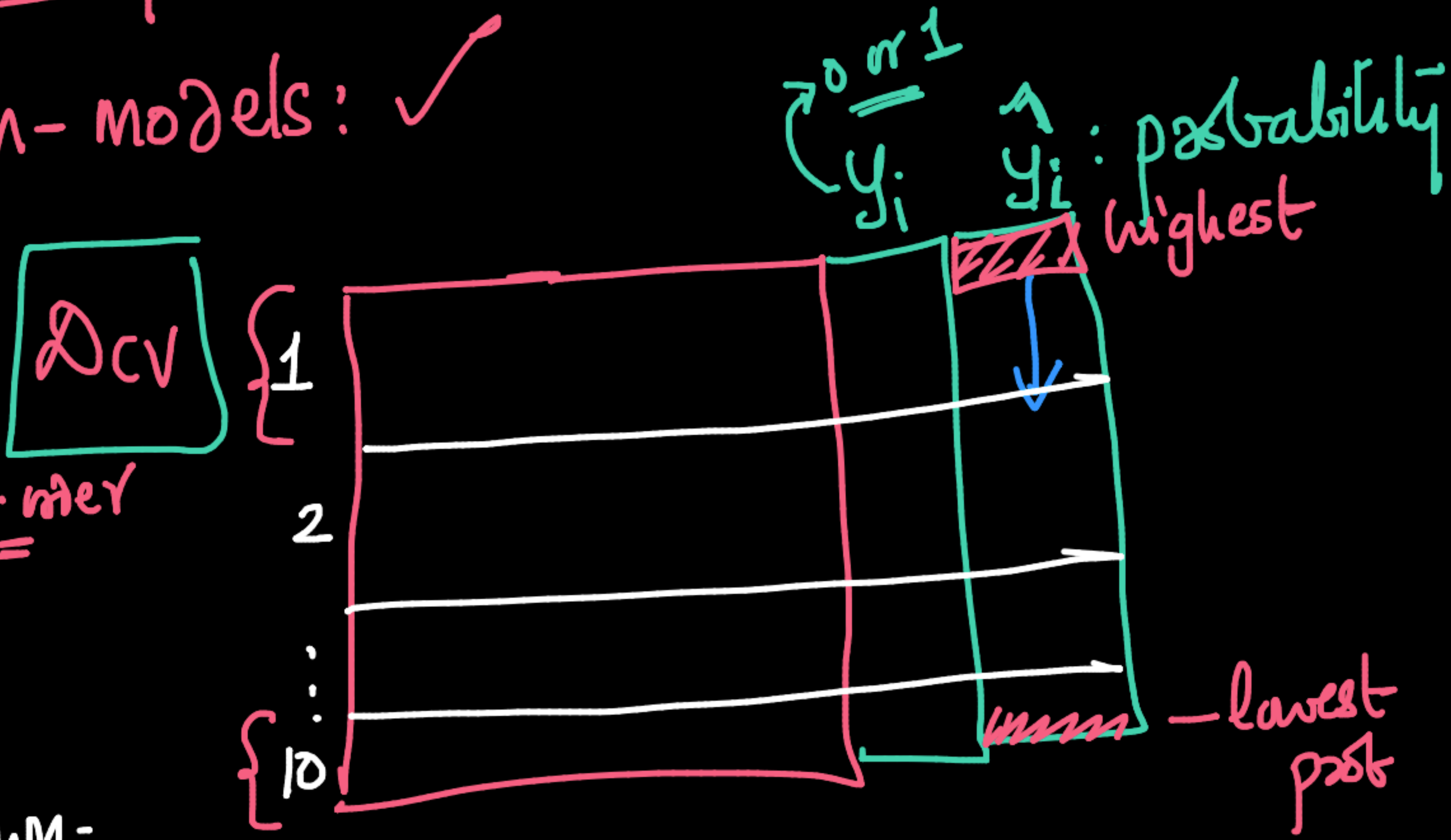
IQR → Univariate

RANSAC → remove outliers based on a robust Model

Lift & gain Charts

Classfn-models: ✓

1. $\hat{y}_i$'s on D_{cv}
2. Sort by $\hat{y}_i$'s in decr. order
3. deciles
4. Build a table as shown.



DCV

$n = 25{,}000$

y_i's

#pts

1st-decile 2179 pts have $y_i = 1$

Decile	Number of Cases	Number of Responses	Cumulative Responses	% of events	Gain	Cumulative Lift
	Input Values					
1	2500	2179	2179	44.71	44.71	4.47
2	2500	1753	3932	35.97	80.67	4.03
3	2500	396	4328	8.12	88.80	2.96
4	2500	111	4439	2.28	91.08	2.28
5	2500	110	4549	2.26	93.33	1.87
6	2500	85	4634	1.74	95.08	1.58
7	2500	67	4701	1.37	96.45	1.38
8	2500	69	4770	1.42	97.87	1.22
9	2500	49	4819	1.01	98.87	1.10
10	2500	55	4874	1.13	100.00	1.00
	25000	4874				

$\frac{44.71}{10}$

$\frac{80.67}{20}$

$\frac{93.33}{50.0}$

|Dcv|

#pts with $y_i = 1$

Gain for i^{th} decile = %age of +ve pts are in i^{th} or smaller decile

Gain

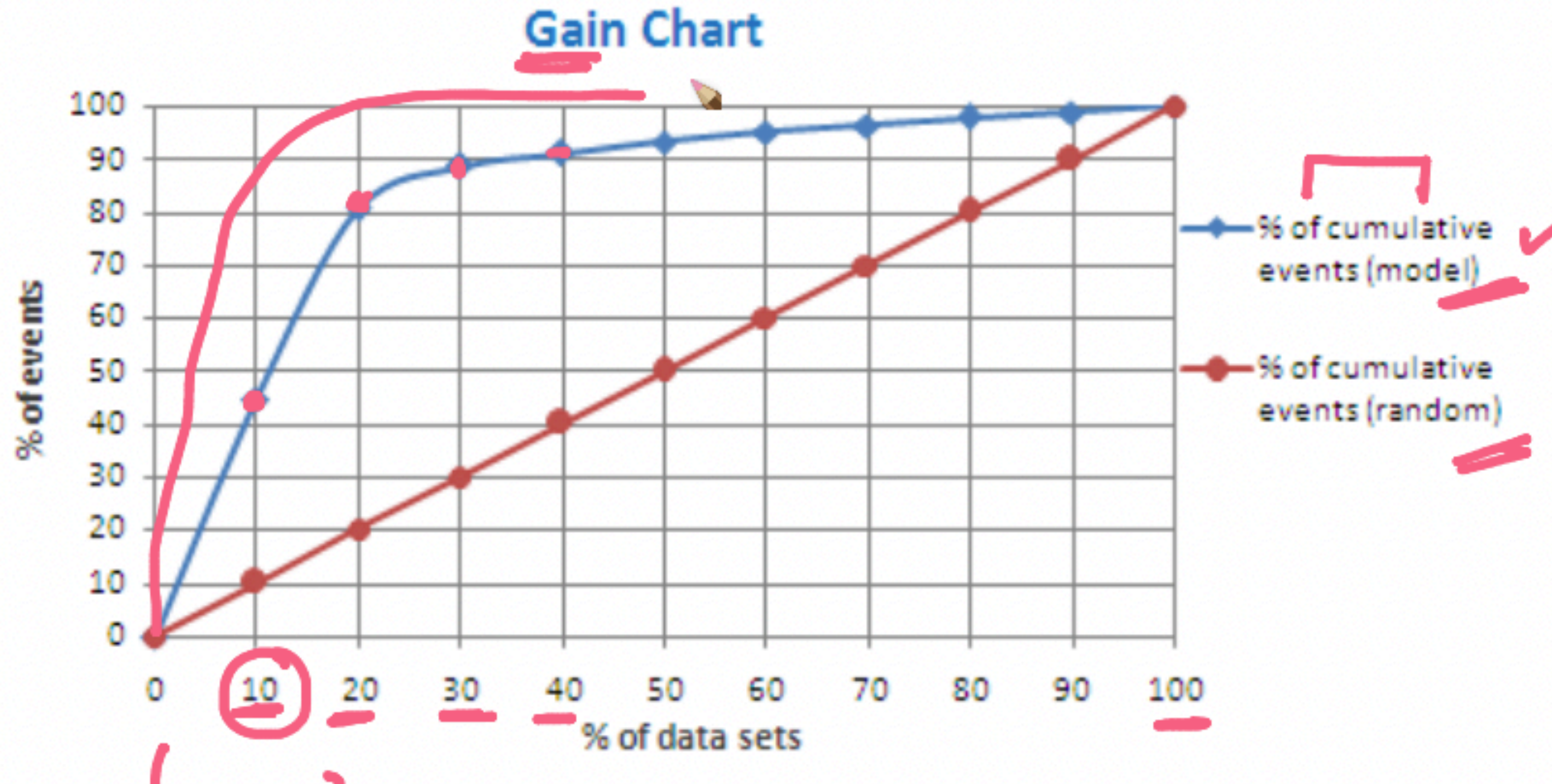
in ith decile = what %age of +ve pts are in ith or smaller decile

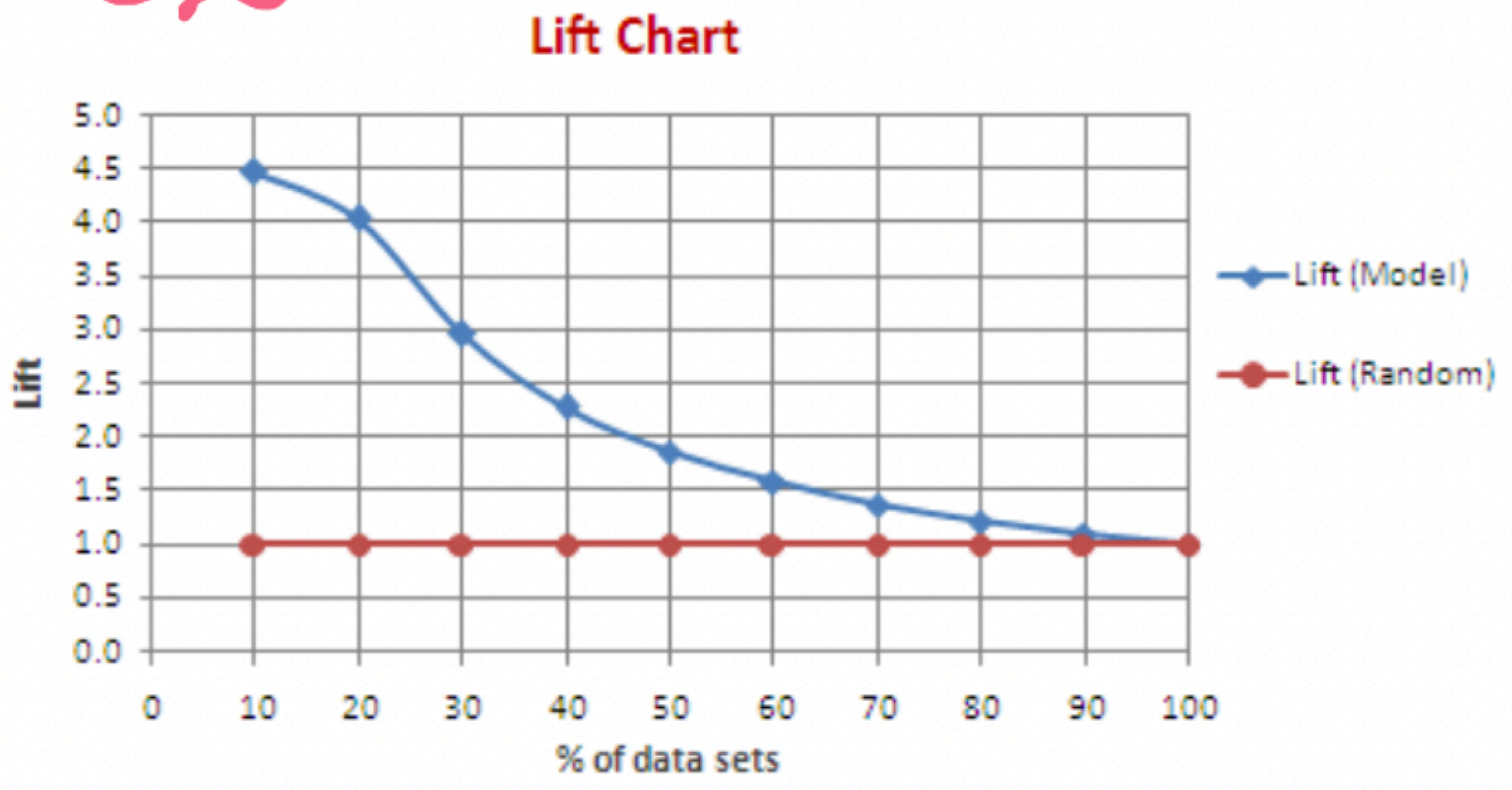
	Input Values					
Decile	Number of Cases	Number of Responses	Cumulative Responses	% of events	Gain	Cumulative Lift
1	2500	2179	2179	44.71	44.71	4.47
2	2500	1753	3932	35.97	80.67	4.03
3	2500	396	4328	8.12	88.80	2.96
4	2500	111	4439	2.28	91.08	2.28
5	2500	110	4549	2.26	93.33	1.87
6	2500	85	4634	1.74	95.08	1.58
7	2500	67	4701	1.37	96.45	1.38
8	2500	69	4770	1.42	97.87	1.22
9	2500	49	4819	1.01	98.87	1.10
10	2500	55	4874	1.13	100.00	1.00
	25000	4874				

cum +ve %age of pts till ith decile

cum %age of +ve pts by random model

	Input Values					
Decile	Number of Cases	Number of Responses	Cumulative Responses	% of events	Gain	Cumulative Lift
1	2500	2179	2179	44.71	44.71	4.47
2	2500	1753	3932	35.97	80.67	4.03

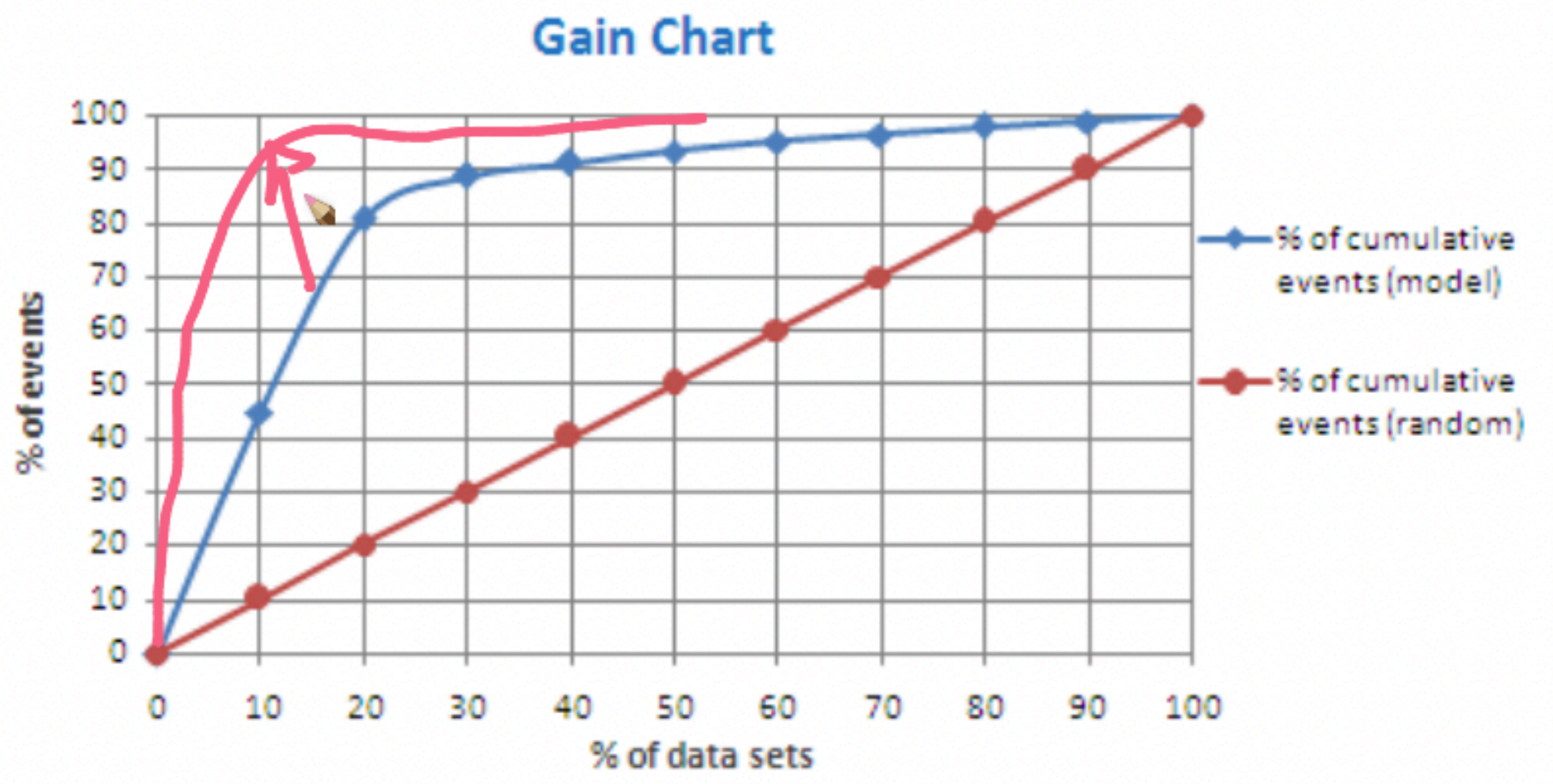


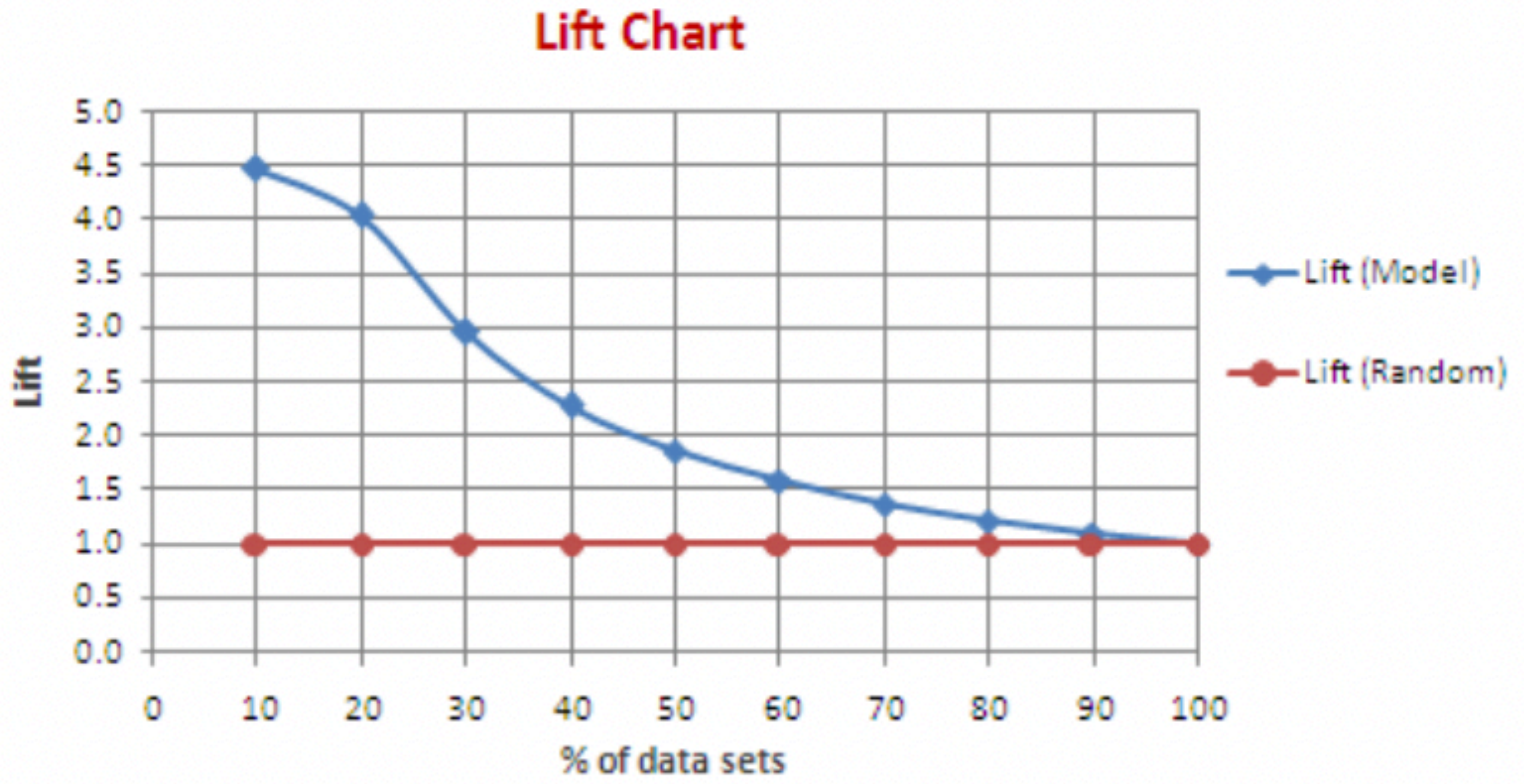


	Input Value					
Decile	Number of Cases	Nu... Re...				
1	2500					
2	2500	1753	3932	35.97	80.67	4.03

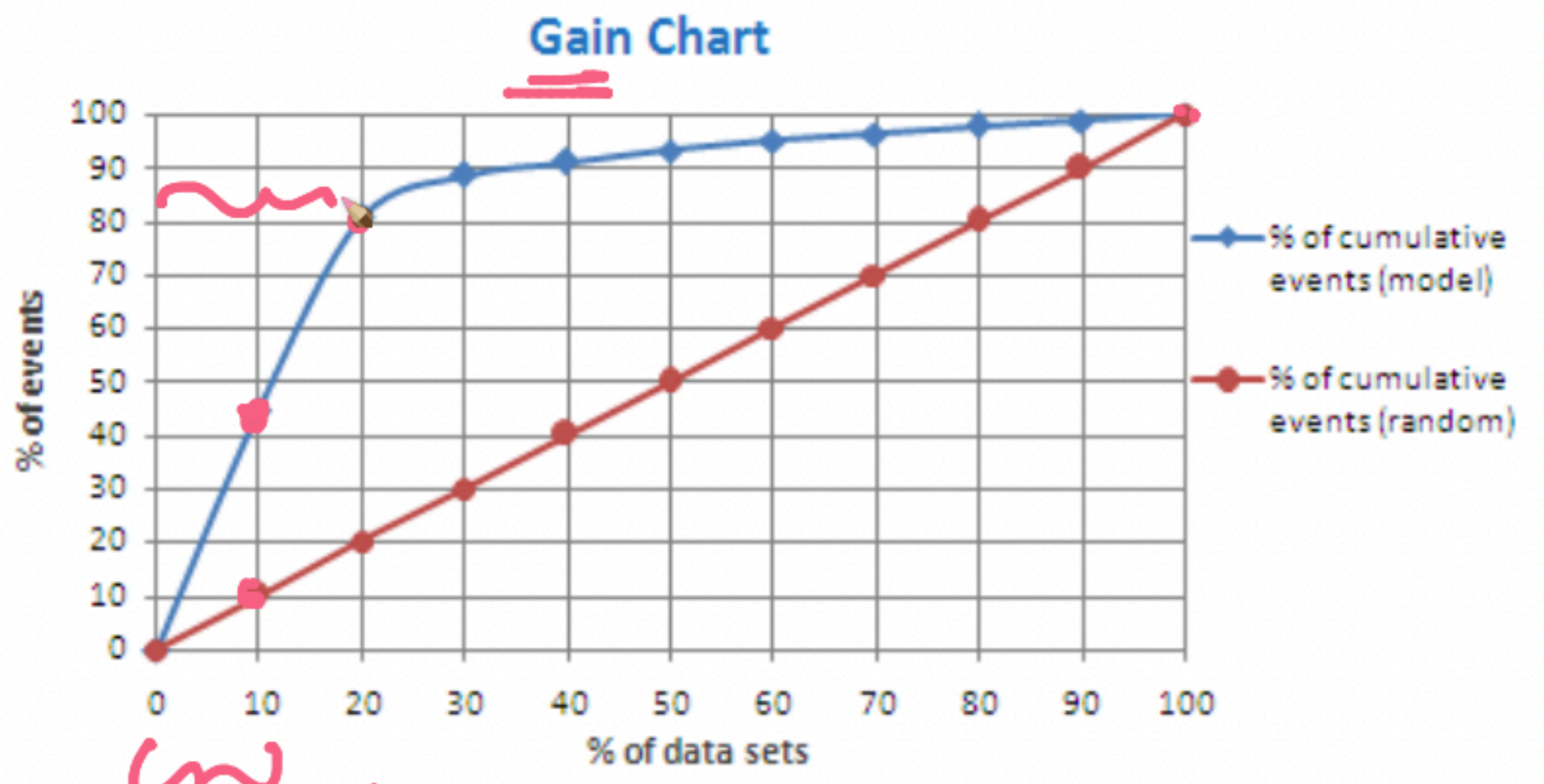
2 of 2

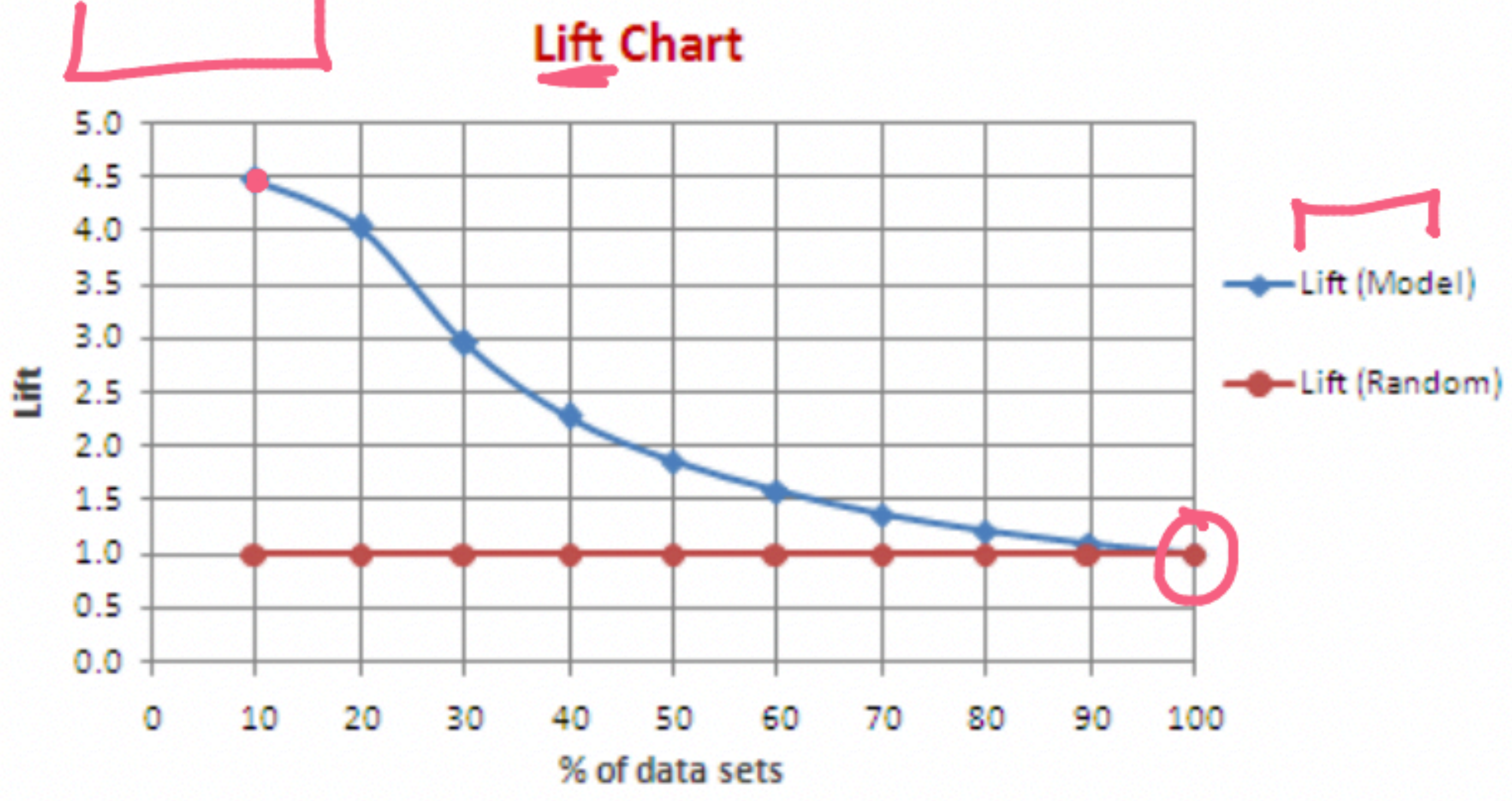
47 / 47





	Input Value					
Decile	Number of Cases	Nu Re				
1	2500					
2	2500	1753	3932	35.97	80.67	4.03





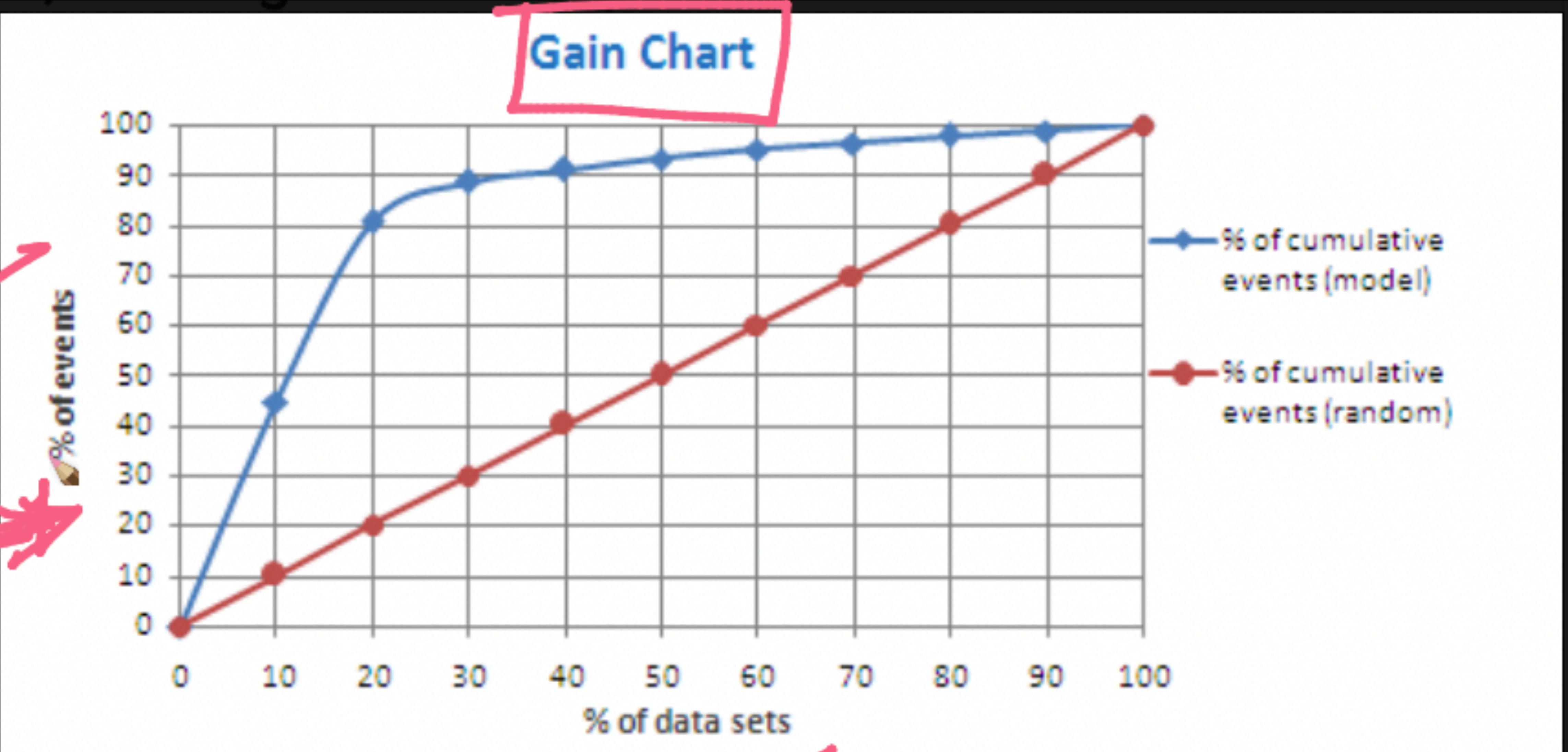
AUC X

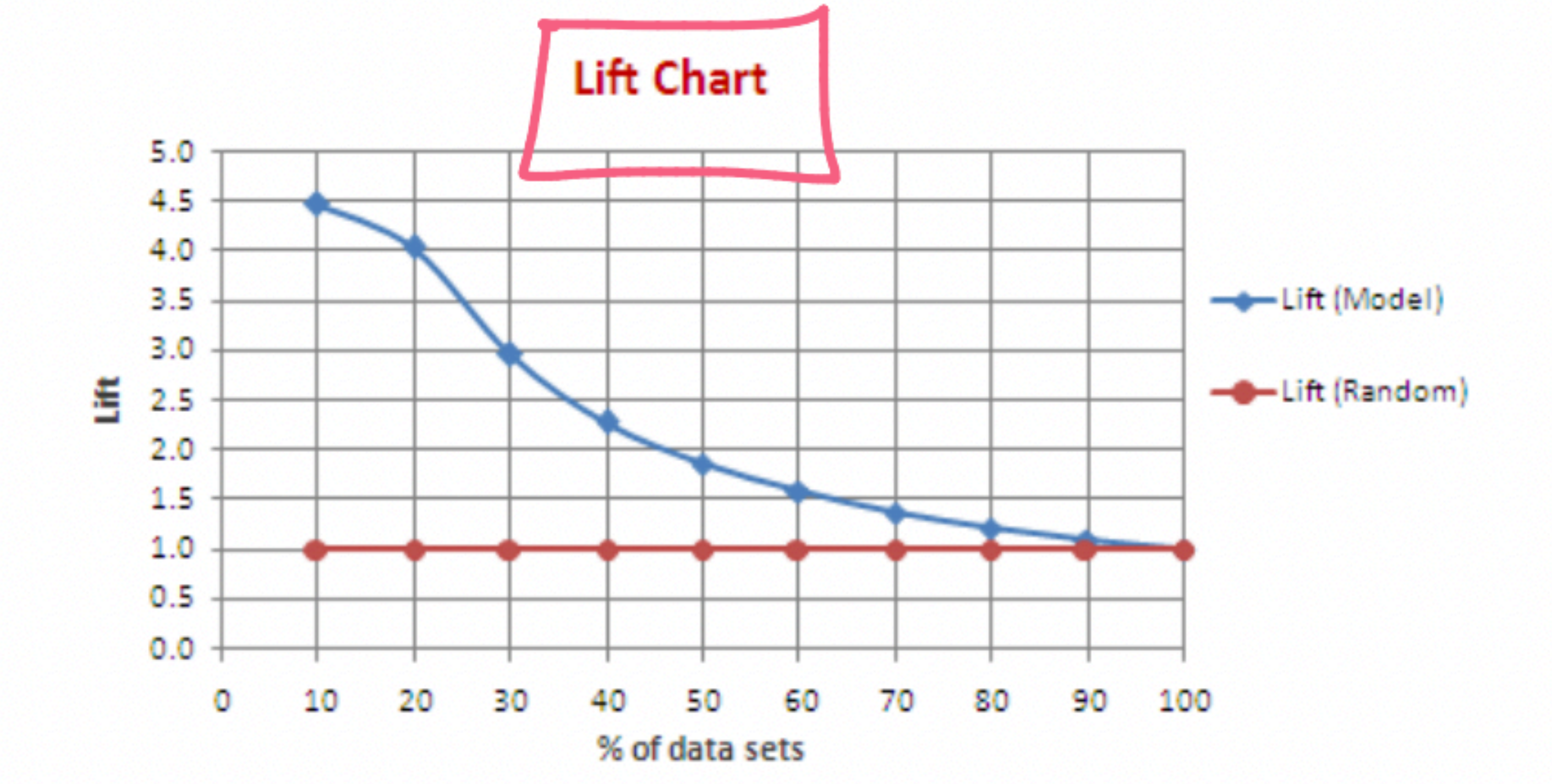
Decile	Input Value					
	Number of Cases	Nu... Re...				
1	2500					
2	2500	1753	3932	35.97	80.67	4.03

Safari File Edit View History Bookmarks Develop Window Help 11:27

www.listendata.com/2014/08/excel-template-gain-and-lift-charts.html

sklearn.linear_... | Probability Cali... | Platt scaling -... | Isotonic regress... | sklearn.isotonic... | Random sample... | Understand Gai... | Metrics Module... | regression - Wi... | Machine Learni... | Regression: A B... | k-means++ - W...

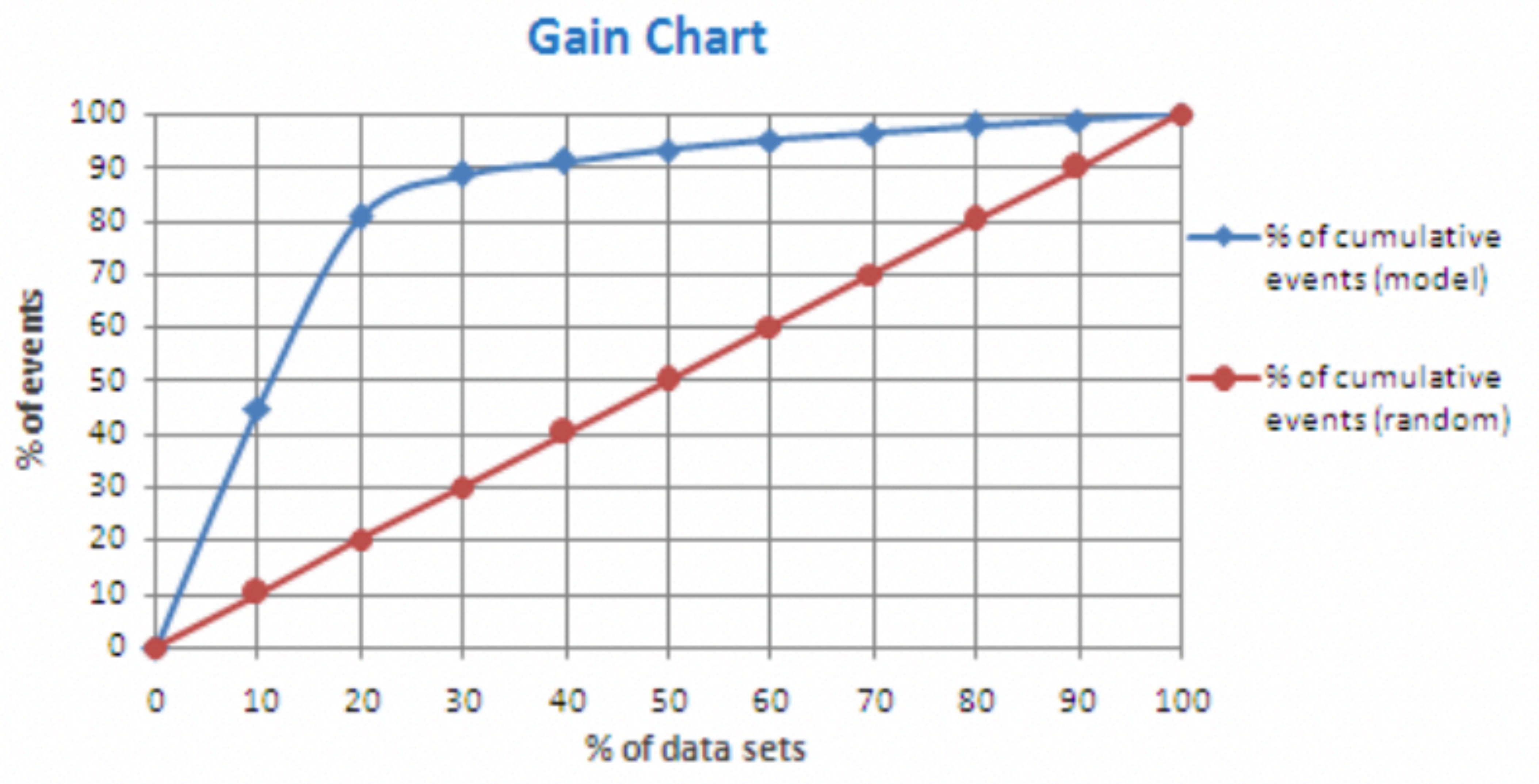


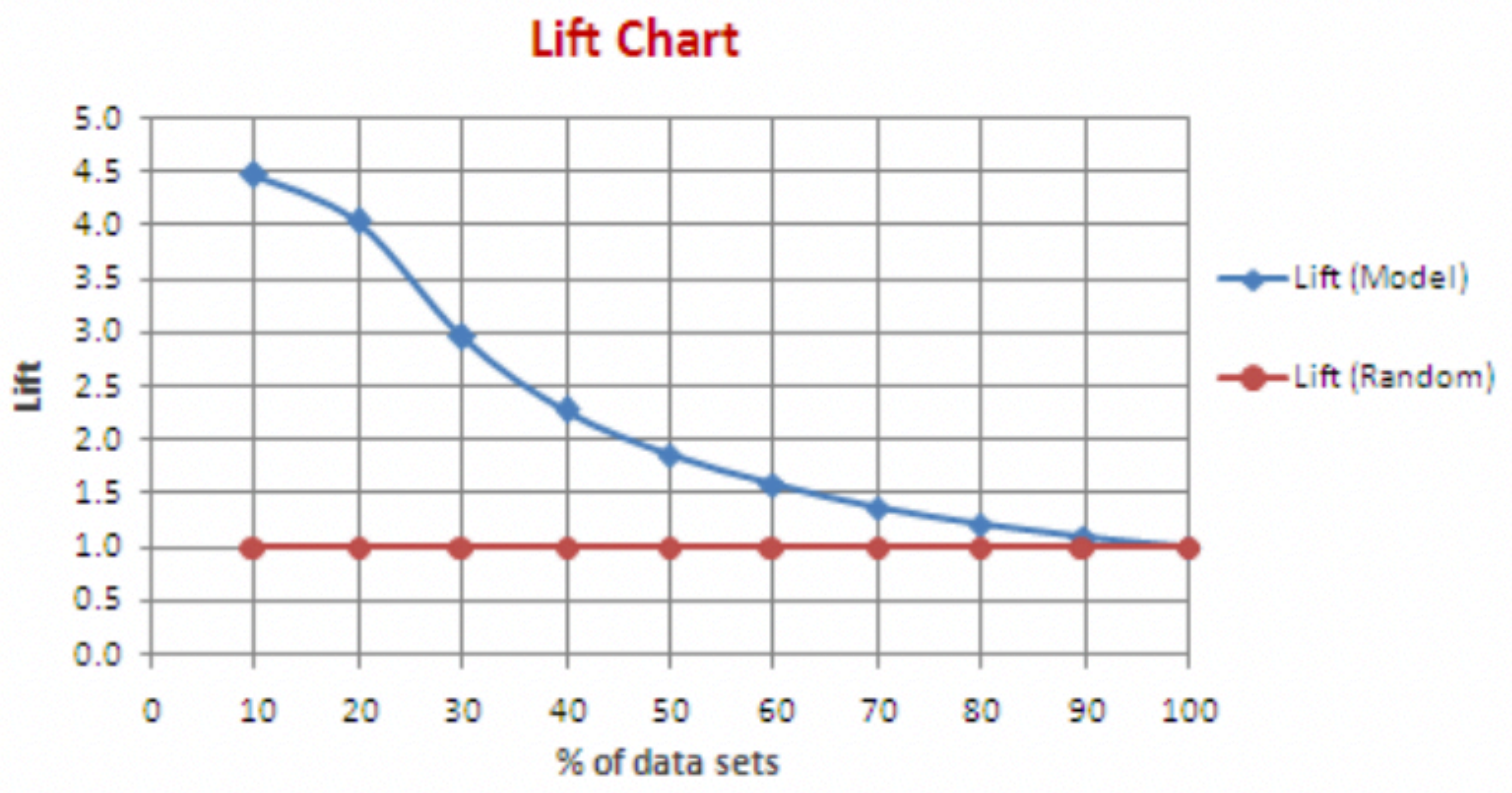


	Input Value					
Decile	Number of Cases	Nu Re				
1	2500					
2	2500	1753	3932	35.97	80.67	4.03

2 of 2

50 / 50


Gain Chart
100
90
80
70
60
50
40
30
20
10
0
% of events
0 10 20 30 40 50 60 70 80 90 100
% of data sets
% of cumulative events (model)
% of cumulative events (random)


Lift Chart
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Lift
0 10 20 30 40 50 60 70 80 90 100
% of data sets
Lift (Model)
Lift (Random)