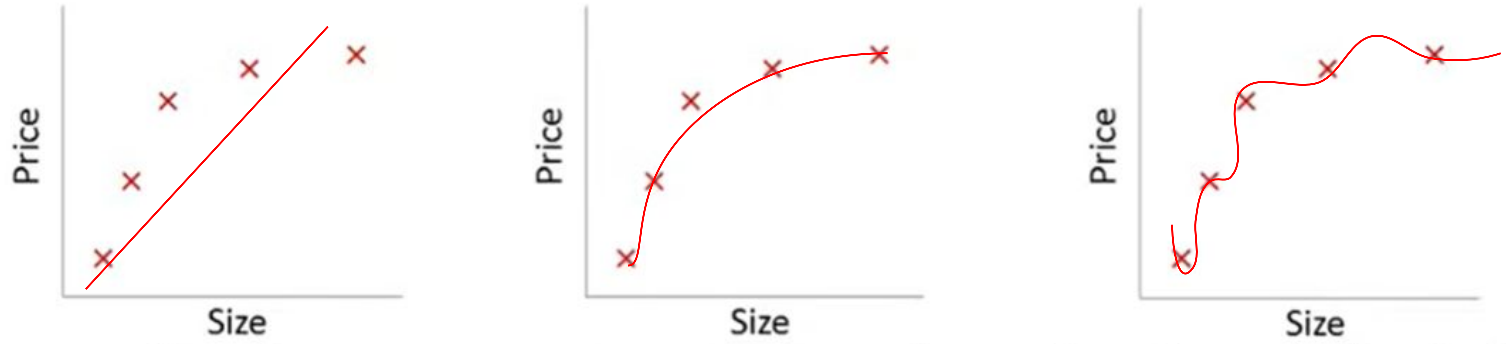
Regularization

source:https://www.youtube.com/watch?v=u73PU6Qwl1I&ab_channel=ArtificialIntelligence-AllinOne

Example: Linear regression (housing prices)

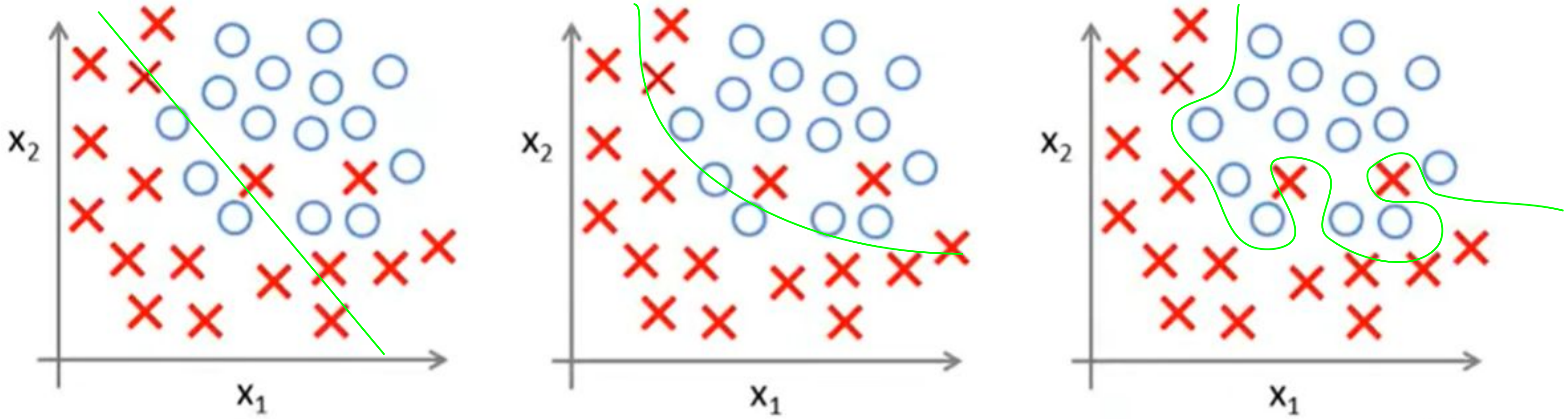


$\theta_0 + \theta_1 x$

$\theta_0 + \theta_1 x + \theta_2 x^2$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Example: Logistic regression



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

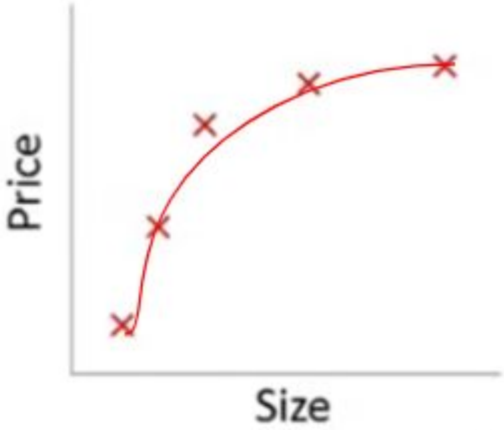
$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2$
$+\theta_3 x_1^2 + \theta_4 x_2^2$
$+\theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2$
$+\theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2$
$+\theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

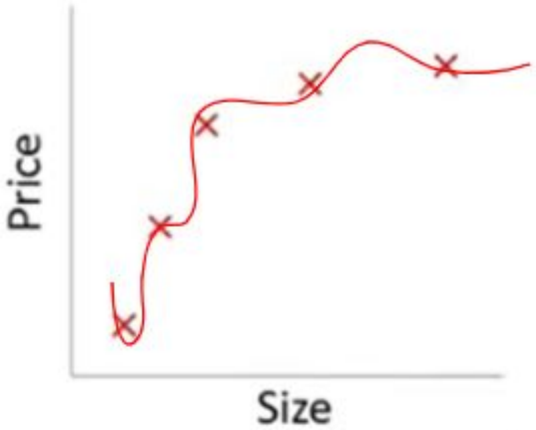
Regularization.

— Keep all the features, but reduce magnitude/values of parameters θ_j.

— Works well when we have a lot of features, each of which contributes a bit to predicting y.



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



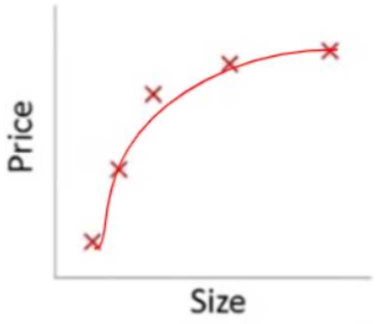
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

In this example if we want to prevent overfit we need to make θ3 and θ4 to be zero.

If our loss function is MSE: we have loss= $\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

If we modify our loss function to be $\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\,\theta_3 + 1000\,\theta_4$

After optimization θ_3 and θ_4 would be zero and we can get



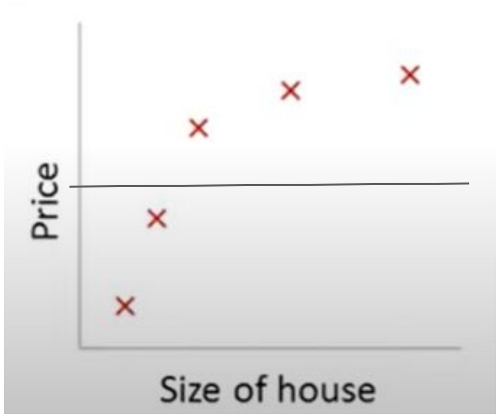
$\theta_0 + \theta_1 x + \theta_2 x^2$

Regularization

Two Regularization:

L1 (Lasso): $$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} |w_i|$$

L2 (Ridge): $$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$

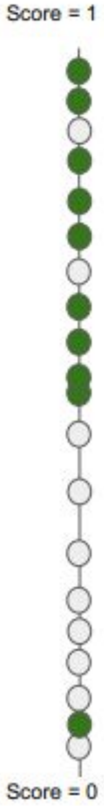


If λ is too big, all weight becomes zero but w0 and we can only get a straight line

Evaluation Metrics

source:https://www.youtube.com/watch?v=Lb1-iNIO
LBw&ab_channel=StanfordOnline

Score based models

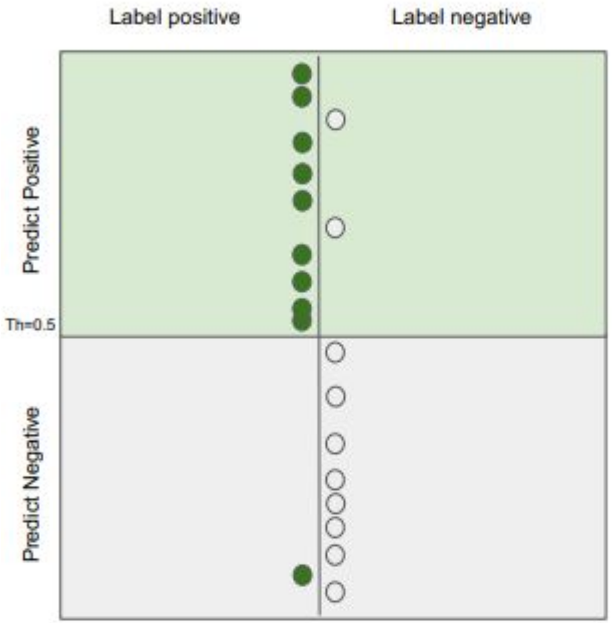


●	Positive example
○	Negative example

Example of Score: Output of logistic regression.
For most metrics: Only ranking matters.
If too many examples: Plot class-wise histogram.

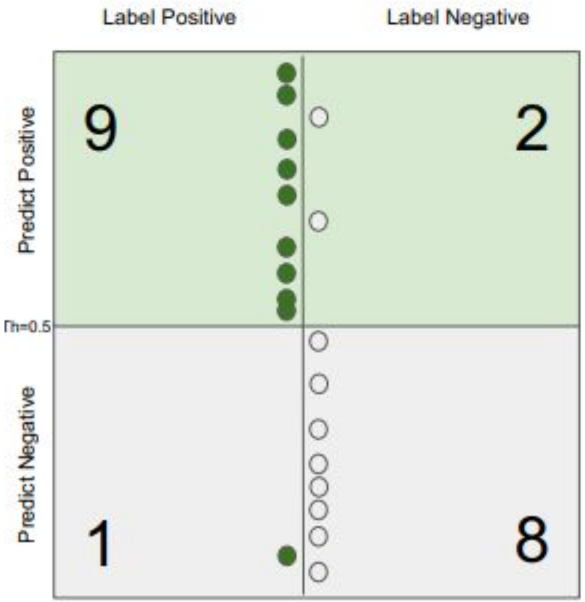
$$\text{Prevalence} = \frac{\text{\# positive examples}}{\text{\# positive examples} + \text{\# negatives examples}}$$

Threshold -> Classifier -> Point Metrics



Th
0.5

Point metrics: Confusion Matrix

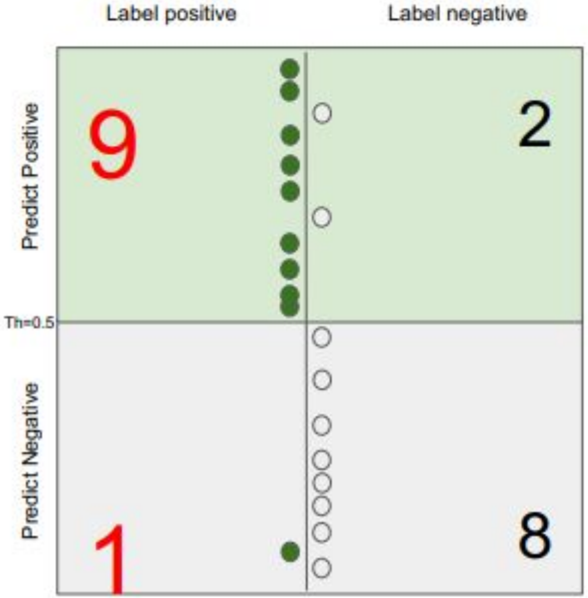


Th	TP	TN	FP	FN
0.5	9	8	2	1

Properties:

- Total sum is fixed (population).
- Column sums are fixed (class-wise population).
- Quality of model & threshold decide how columns are split into rows.
- We want diagonals to be “heavy”, off diagonals to be “light”.

Point metrics: Positive Recall (Sensitivity)

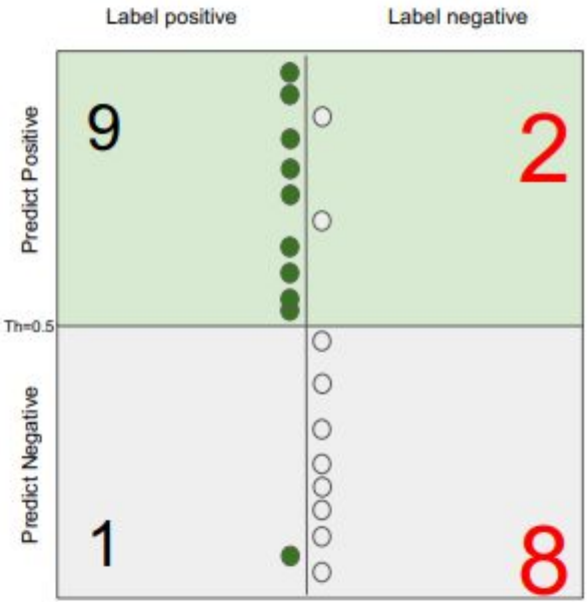


Th	TP	TN	FP	FN	Acc	Pr	Recall
0.5	9	8	2	1	.85	.81	.9

Trivial 100% recall = pull everybody above the threshold.
Trivial 100% precision = push everybody below the threshold except 1 green on top.
(Hopefully no gray above it!)

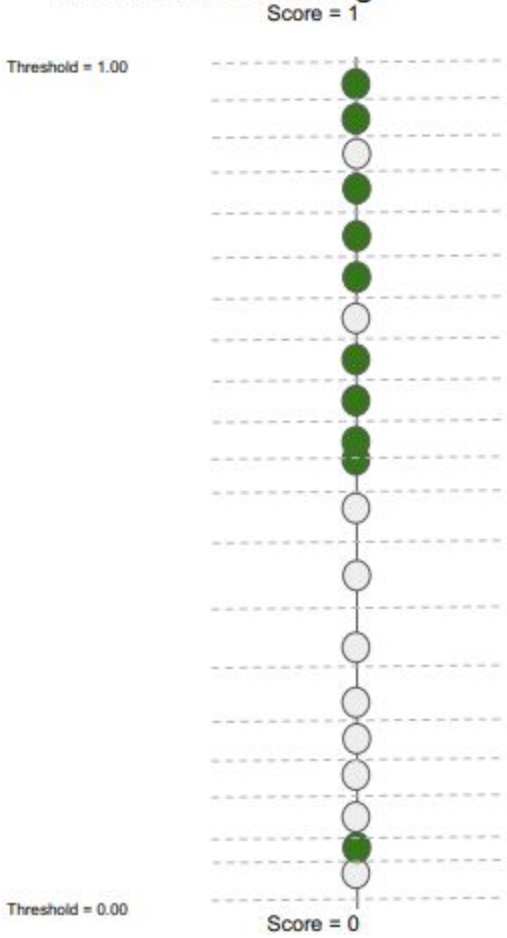
Striving for good precision with 100% recall =
pulling up the lowest green as high as possible in the ranking.
Striving for good recall with 100% precision =
pushing down the top gray as low as possible in the ranking.

Point metrics: Negative Recall (Specificity)



Th	TP	TN	FP	FN	Acc	Pr	Recall	Spec
0.5	9	8	2	1	.85	.81	.9	0.8

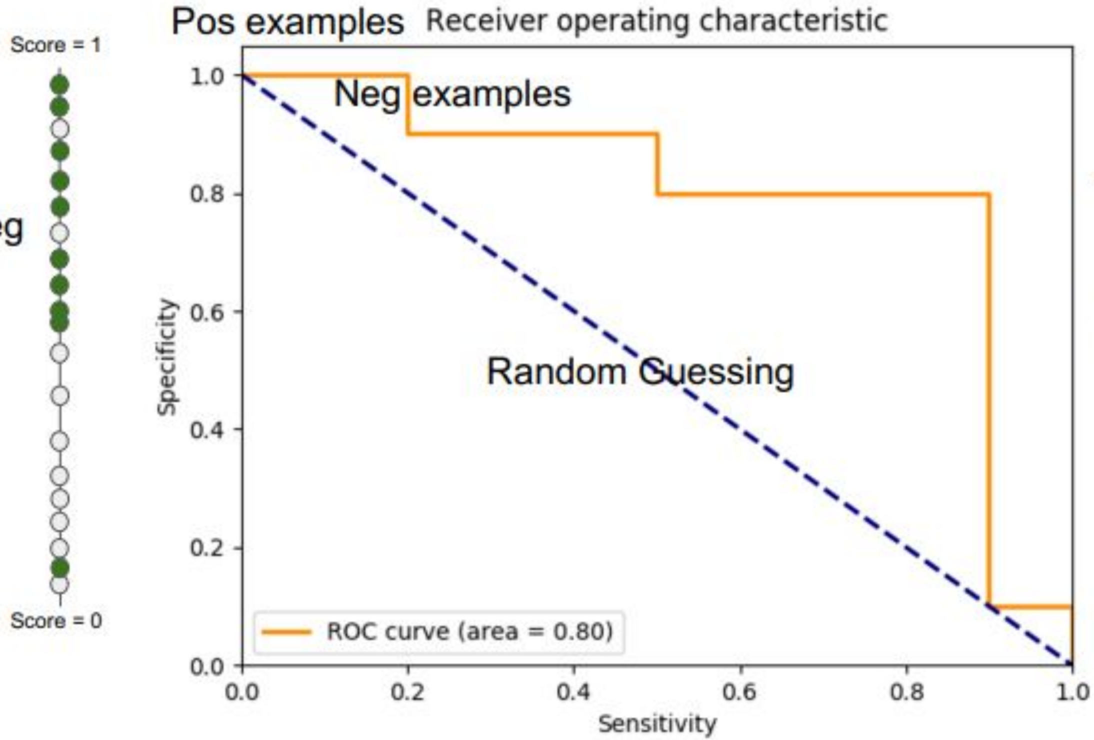
Threshold Scanning



Threshold	TP	TN	FP	FN	Accuracy	Precision	Recall	Specificity	F1
1.00	0	10	0	10	0.50	1	0	1	0
0.95	1	10	0	9	0.55	1	0.1	1	0.182
0.90	2	10	0	8	0.60	1	0.2	1	0.333
0.85	2	9	1	8	0.55	0.667	0.2	0.9	0.308
0.80	3	9	1	7	0.60	0.750	0.3	0.9	0.429
0.75	4	9	1	6	0.65	0.800	0.4	0.9	0.533
0.70	5	9	1	5	0.70	0.833	0.5	0.9	0.625
0.65	5	8	2	5	0.65	0.714	0.5	0.8	0.588
0.60	6	8	2	4	0.70	0.750	0.6	0.8	0.667
0.55	7	8	2	3	0.75	0.778	0.7	0.8	0.737
0.50	8	8	2	2	0.80	0.800	0.8	0.8	0.800
0.45	9	8	2	1	0.85	0.818	0.9	0.8	0.857
0.40	9	7	3	1	0.80	0.750	0.9	0.7	0.818
0.35	9	6	4	1	0.75	0.692	0.9	0.6	0.783
0.30	9	5	5	1	0.70	0.643	0.9	0.5	0.750
0.25	9	4	6	1	0.65	0.600	0.9	0.4	0.720
0.20	9	3	7	1	0.60	0.562	0.9	0.3	0.692
0.15	9	2	8	1	0.55	0.529	0.9	0.2	0.667
0.10	9	1	9	1	0.50	0.500	0.9	0.1	0.643
0.05	10	1	9	0	0.55	0.526	1	0.1	0.690
0.00	10	0	10	0	0.50	0.500	1	0	0.667

Summary metrics: Rotated ROC (Sen vs. Spec)

Specificity
= True Neg / Neg



AUROC = Area Under ROC

= Prob[Random Pos ranked higher than random Neg]

Agnostic to prevalence!

Sensitivity = True Pos / Pos