CS 307

Fall 2023

Dalpiaz

Week 09

LOGISTIC REGRESSION

So far...

ESTIMATING Y/X

Create $C(x)$ using $\hat{p}_k(x)$

$$\hat{p}_k(x) = \hat{P}[Y = k \mid X = x] \approx$$

Proportion of $y_i = k$ "near" x

- KNN NEIGHBORS
- TREE NEIGHBORHOODS

NONPARAMETRIC

Now...

A parametric method for

binary classification

Binary Classification

$$Y = \begin{cases} 1 & \text{"positive"} \\ 0 & \text{"negative"} \end{cases}$$

Define

$$p(x) = P[Y=1 \mid X=x]$$

$$1-p(x) = P[Y=0 \mid X=x]$$

LOGISTIC REGRESSION

$$\log\left(\underbrace{\frac{p(x)}{1-p(x)}}_{\text{ODDS}}\right) = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p}_{\text{LINEAR COMBO OF FEATURES}}$$

$$p(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

$f(x, \beta)$

KNOW

LEARN

LOGISTIC REGRESSION

$$Y \mid X \sim \text{BERN}(\underbrace{p}_{f(x,\beta)}(x))$$

COMPARE TO ORDINARY LINEAR REGRESSION

$$Y \mid X \sim N(\underbrace{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}_{\text{LINEAR COMBO}}, \sigma^2)$$

PARAMETER

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

$$\varepsilon \sim N(0, \sigma^2)$$

DEFINE

$$\text{logit}(z) = \log\left(\frac{z}{1-z}\right)$$

SOME INPUT

$$\sigma(z) = \text{logit}^{-1}(z) = \frac{e^{z}}{1+e^{z}} = \frac{1}{1+e^{-z}}$$

SIGMOID FUNCTION

INVERSE LOGIT

$$\eta(x) = \beta_0 + \beta_1 x_1 + \cdots \beta_p x_p$$

$$\text{logit} : [0,1] \to \mathbb{R}$$

$$\sigma : \mathbb{R} \to [0,1]$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

$$\boxed{\text{logit}(p(x)) = \eta(x)}$$

$$p(x) = \sigma(\eta(x)) = \frac{e^{\eta(x)}}{1+e^{\eta(x)}} = \frac{e^{\beta_0+\beta_1 x_1 + \cdots + \beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1 + \cdots + \beta_p x_p}}$$

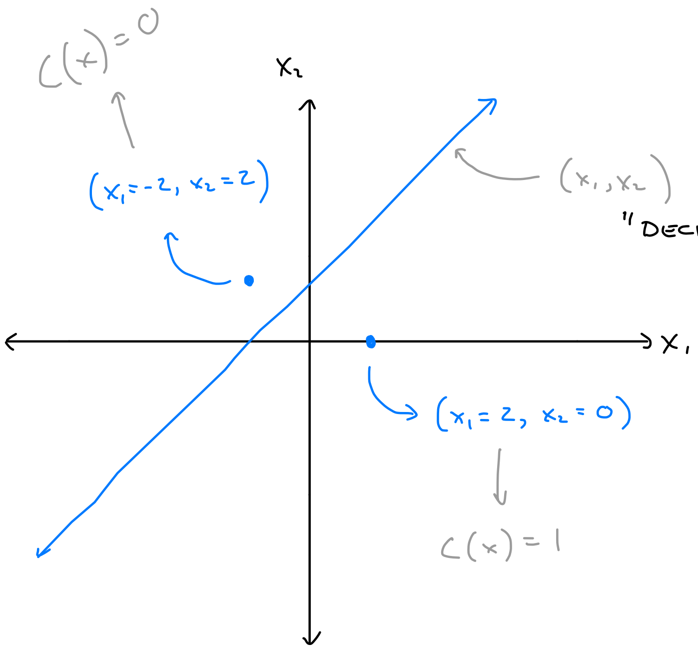
Example

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \underset{\beta_0}{4} + \underset{\beta_1}{2}x_1 - \underset{\beta_2}{2}x_2$$

Note $p(x) = 0.5 \iff \eta(x) = 0$

$$0 = 4 + 2x_1 - 2x_2 = \eta(x)$$

$$\Rightarrow \quad x_2 = 2 + x_1$$



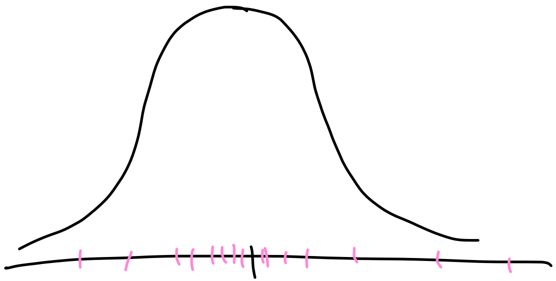
$(x_1, x_2) : p(x) = 0.5$

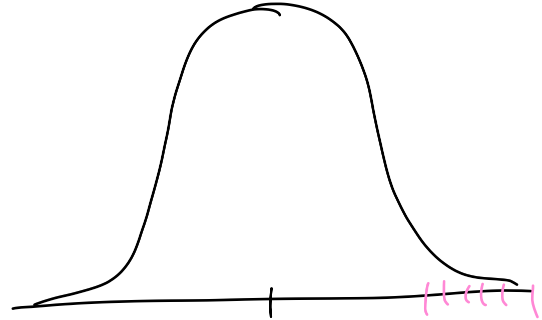
"Decision Boundary"

$$P(x_1 = 2, x_2 = 0) = \frac{1}{1 + e^{-(4+4+0)}} = 0.9996$$

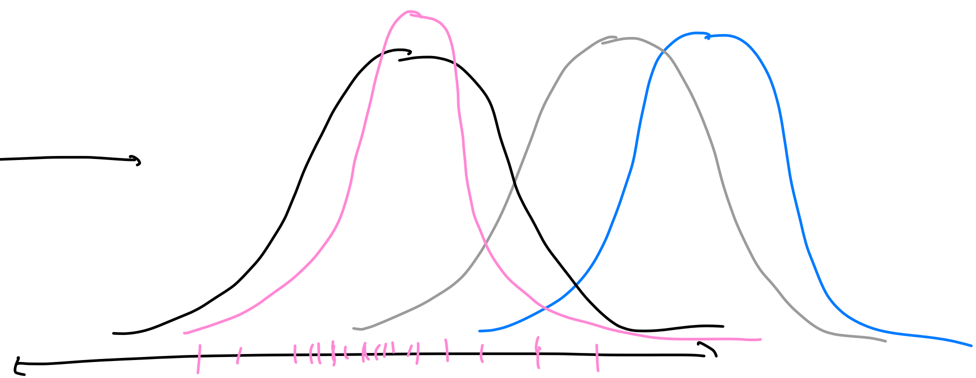
$$P(x_1 = -2, x_2 = 2) = \frac{1}{1 + e^{-(4-4-4)}} = 0.01799$$

A NOTE ABOUT LIKELIHOOD





μ, σ^2



FITTING LOGISTIC TO DATA

x_i	y_i	$p(x_i)$
2	1	
3	1	
1	1	
3	1	
5	1	
4	0	
5	0	
6	0	
7	0	
6	0	

SHOULD BE "LARGE"

SHOULD BE "SMALL"

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

$y_1 \quad y_2 \quad y_3$

SEQUENCE : 1, 1, 0

PROBABILITY : $p(x_1) \cdot p(x_2) \cdot (1-p(x_3))$

CONDITIONAL LIKELIHOOD

$$\mathcal{L}(\beta_0, \beta_1) = \prod_{i=1}^{n} \underbrace{P[Y_i = y_i \mid X_i = x_i]}_{f(x, \beta)}$$

MAXIMIZE

KNOWN

LEARN

LIKELIHOOD

$$\mathcal{L}(\beta_0, \beta_1) = \prod_{i=1}^{n} P[Y_i = y_i \mid X_i = x_i] = \prod_{i=1}^{n} p(x_i)^{y_i} \left(1 - p(x_i)\right)^{1-y_i}$$

$y_i = 1$

$1 - y_i = 0$

$$\boxed{\log \mathcal{L}(\beta_0, \beta_1)} = \underbrace{\sum_{i=1}^{n} y_i \log(p(x_i))}_{\text{CLASS 1}} + \underbrace{\sum_{i=1}^{n} (1-y_i) \log(1 - p(x_i))}_{\text{CLASS 0}}$$

LOG-LIKELIHOOD

$$= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log\left(\frac{p(x_i)}{1 - p(x_i)}\right)$$

$$= \sum_{i=1}^{n} \log\left(1 - \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}\right) + \sum_{i=1}^{n} y_i (\beta_0 + \beta_1 x_i)$$

SHOWS βS

$$= -\sum_{i=1}^{n} \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right) + \sum_{i=1}^{n} y_i (\beta_0 + \beta_1 x_i)$$

$$\log \mathcal{L}(\beta_0, \beta_1) = -\sum_{i=1}^{n} \log\left(1 + e^{\beta_0 + \beta_1 x_i}\right) + \sum_{i=1}^{n} y_i (\beta_0 + \beta_1 x_i)$$

$$\boxed{\begin{aligned} \frac{d}{d\beta_0} \log \mathcal{L}(\beta_0, \beta_1) &= -\sum_{i=1}^{n} \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} + \sum_{i=1}^{n} y_i = 0 \\ \frac{d}{d\beta_1} \log \mathcal{L}(\beta_0, \beta_1) &= -\sum_{i=1}^{n} x_i \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} + \sum_{i=1}^{n} x_i y_i = 0 \end{aligned}}$$

- NO CLOSED FORM SOLUTION
- USE NUMERIC OPTIMIZATION
  - NEWTON'S METHOD
  - ETC

OR...

Logistic Regression in Python

sklearn.linear_model.LogisticRegression

- .fit
- .predict
- .predict_proba

NOT LOGISTIC REGRESSION BY DEFAULT...

MUST SET

penalty = None

PENALIZED LOGISTIC REGRESSION

ERROR + λ PENALTY

NEGATIVE LOG-LIKELIHOOD

ℓ_1 "LASSO" $\sum_{i=1}^{p} |B_i|$

penalty = "L1"

ℓ_2 "RIDGE" $\sum_{i=1}^{p} B_i^2$

penalty = "L2"

BINARY CLASSIFICATION

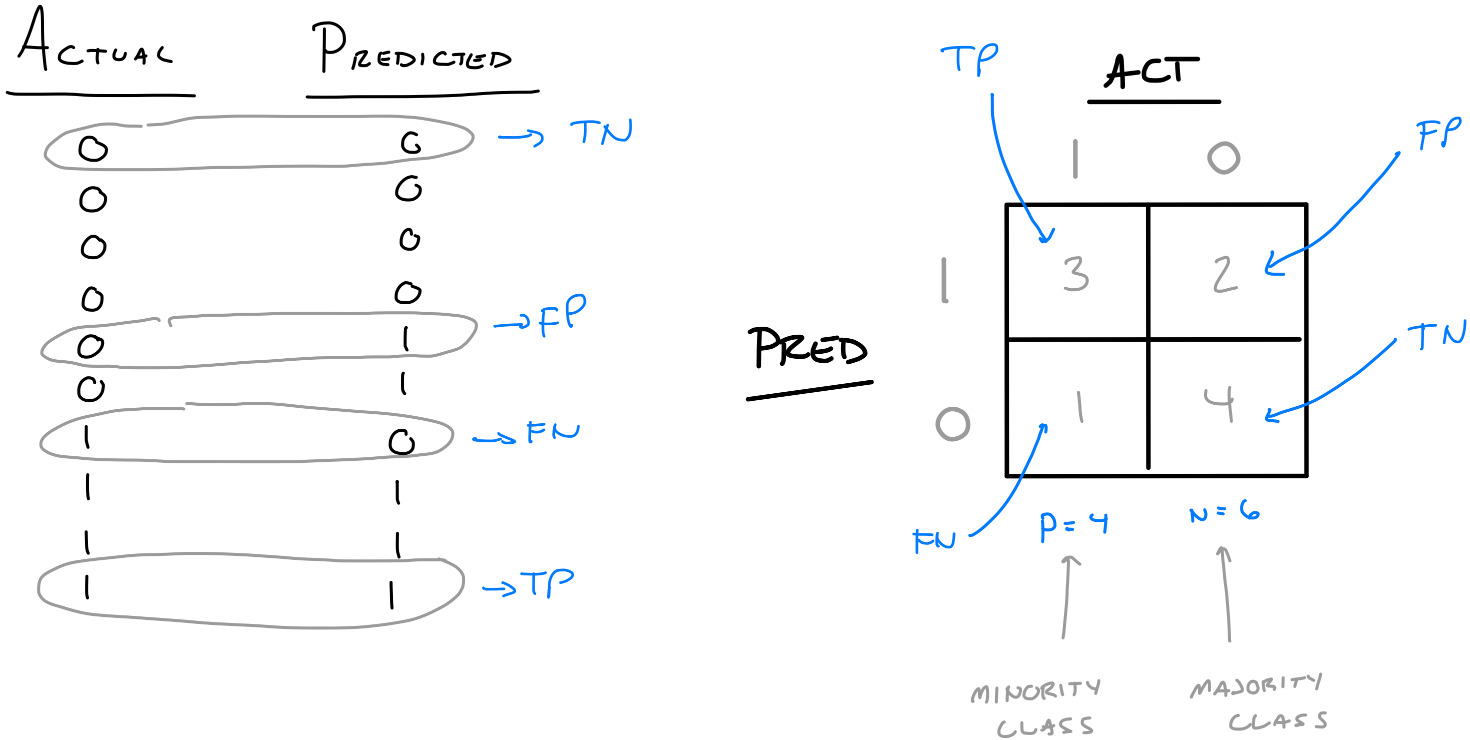
(x_i, y_i) FROM
- → VALIDATION → EVAL
  - ↳ MODEL FIT TO VTRAIN
- OR
- → TEST → REPORT
  - ↳ MODEL FIT TO TRAIN

y_i ↓ ACTUAL	$C(x_i)$ ↓ PREDICTED	
0	0	
0	0	
0	0	
0	0	
0	1	→ "POSITIVE"
0	1	
1	0	→ "NEGATIVE"
1	1	
1	1	
1	1	

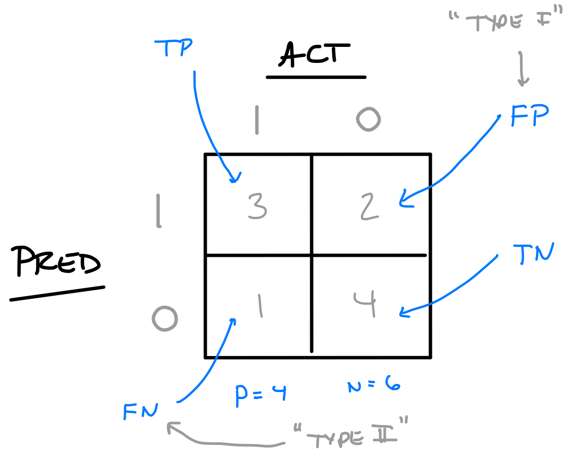
CONFUSION MATRIX

$$
\begin{array}{cc|c|c|}
 & & \multicolumn{2}{c}{\text{ACT}} \\
 & & 1 & 0 \\
\hline
\text{PRED} & 1 & ||| & || \\
\hline
 & 0 & | & |||| \\
\hline
\end{array}
$$

↑ POSITIONS OF ACT / PRED AND 0/1 COULD CHANGE!!!



ACTUAL
PREDICTED
0 0 → TN
0 0
0 0
0 0
0 1 → FP
0 1
1 0 → FN
1 1
1 1
1 1 → TP
TP
ACT
FP
1
0
PRED
1
3
2
TN
0
1
4
FN
P = 4
N = 6
MINORITY CLASS
MAJORITY CLASS



1 − MISCLASS

$$* \ Acc = \frac{TP + TN}{P + N} = \frac{3+4}{4+6} = 0.7$$

"ALONG DIAGONAL"

$$\text{PREVALANCE} = \frac{P}{P+N} = \frac{4}{10} = 0.4$$

$1 - 0.4 = 0.60$

RECALL

SENSITIVITY

$$TPR = \frac{TP}{TP+FN} = \frac{TP}{P} = \frac{3}{4} = 0.75$$

$$FPR = \frac{FP}{FP+TN} = \frac{FP}{N} = \frac{2}{6} = 0.3\overline{3}$$

SPECIFICITY

$$TNR = \frac{TN}{FP+TN} = \frac{TN}{N} = \frac{4}{6} = 0.6\overline{6}$$

PRECISION

$$PPV = \frac{TP}{PP} = \frac{3}{5} = 0.60$$

$$FDR = \frac{FP}{PP} = \frac{2}{5} = 0.40$$

BALANCED ACCURACY

$$BA = \frac{TPR + TNR}{2} = \frac{0.75 + 0.6\overline{6}}{2}$$

$$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}$$

$$\text{Recall} = \frac{TP}{P} \approx P(\hat{y}=1 \mid y=1)$$

$$\text{Precision} = \frac{TP}{PP} \approx P(y=1 \mid \hat{y}=1)$$

$$\frac{P(y=1 \cap \hat{y}=1)}{P(\hat{y}=1)}$$

TP

Predicted positives

FP versus FN

WHICH IS WORSE?

↳ CONSIDER CONTEXT!

MEDICAL TESTING
- FP → DO FURTHER TESTING
- FN → UNAWARE OF DISEASE!

SPAM DETECTION
- FP → DON'T SEE IMPORTANT EMAIL!
- FN → NEED TO DELETE SPAM FROM INBOX

NO INFORMATION RATE

$\longrightarrow$ PROPORTION OF MAJORITY CLASS

$$NIR = \max\left\{ \underbrace{\frac{P}{P+N}}_{\text{PREV}} , \underbrace{\frac{N}{P+N}}_{1-\text{PREV}} \right\}$$

IF ACC > NIR $\longrightarrow$ REASONABLE CLASSIFIER

IF ACC < NIR $\longrightarrow$ USELESS CLASSIFIER

0.7 > 0.6 ✓

$Y = 0$ OR $Y = 1$

$$p(x) \triangleq P[Y = 1 \mid X = x]$$

$$1 - p(x) = P[Y = 0 \mid X = x]$$

$$P[Y = 1 \mid X = x] \geq P[Y = 0 \mid X = x]$$

$$C^B(x) = \begin{cases} 1 & p(x) \geq 0.5 \\ 0 & p(x) < 0.5 \end{cases}$$

ESTIMATE $p(x)$ USING
- KNN
- TREE
- LOGISTIC
- NAIVE BAYES

$\hat{p}(x)$

"THRESHOLD"

SET $0 \le \alpha \le 1$

$$C_\alpha(x) = \begin{cases} 1 & \hat{p}(x) \ge \alpha \\ 0 & \hat{p}(x) < \alpha \end{cases}$$

ASSUMED $\alpha = 0.5$

AS $\alpha \uparrow$ HARDER TO CLASSIFY AS $Y=1$

CLASSIFIER $= f(\hat{p}(x), \alpha)$

y_i	$\hat{p}(x_i)$	$C_{0.0}(x_i)$	$C_{0.25}(x_i)$	$C_{0.5}(x_i)$	$C_{0.75}(x_i)$	$C_{1.0}(x_i)$
0	0.1	1	0	0	0	0
0	0.1	1	0	0	0	0
0	0.2	1	0	0	0	0
0	0.3	1	1	0	0	0
0	0.6	1	1	1	0	0
0	0.7	1	1	1	0	0
1	0.4	1	1	0	0	0
1	0.7	1	1	1	0	0
1	0.8	1	1	1	1	0
1	0.9	1	1	1	1	0

		$C_{0.0}$	$C_{0.25}$	$C_{0.5}$	$C_{0.75}$	$C_{1.0}$
TP/P =	SENS (TPR) =	1.00	1.00	0.75	0.50	0.00
TN/N =	SPEC (1-FPR) =	0.00	0.50	0.66	1.00	1.00
	ACC =	0.40	0.70	0.70	0.80	0.60

EXPECTED TO BE BEST

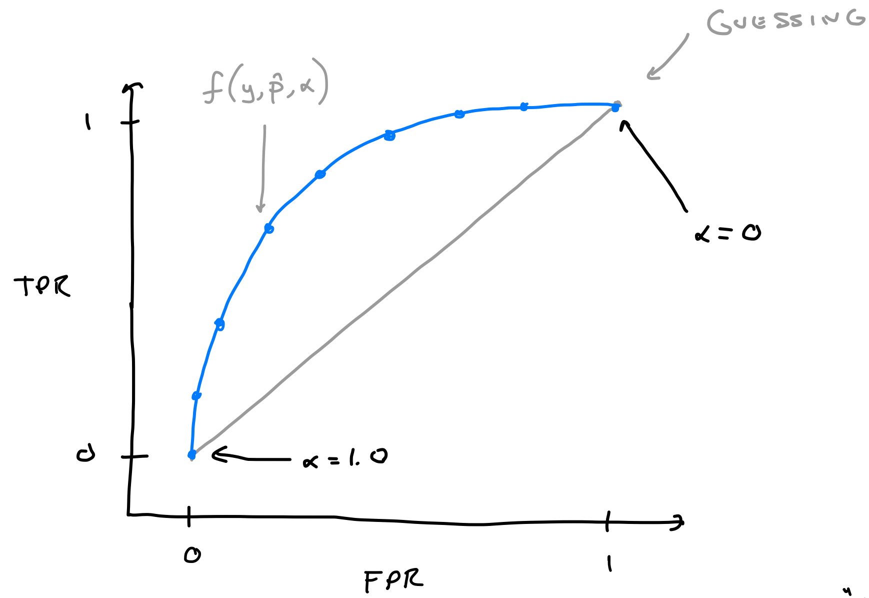
Evaluate $\hat{p}(x)$ instead of $C(x)$? ROC Curve!

<u>Inputs</u>

y

$\hat{p}(x)$

αs



AUC → "Area under curve"

- Bigger = Better
- 1 = Perfect
- 0.50 = "Worst"

Also Precision-Recall

WHAT IF DATA IS IMBALANCED?

CARE ABOUT TPR

$Y = 1$ VERY RARE

$Y = 0$ EXTREMELY COMMON

EASY TO GET GOOD ACCURACY

↳ JUST PREDICT $Y = 0$ ALWAYS!

TPR
BA
F_1
ROC_AUC

NEED OTHER METRICS

+ MODEL ADJUSTMENTS

Do Nothing	Easy!	Worth trying!!!
Change Threshold	Hard to validate!	Reduces acc
Up / Down sample (SMOTE)	Tricky!	Reduces acc
Specify class_weights	Good!	Reduces acc
Use class_weights = "balanced"	Magic!	Reduces acc