Classification and Logistic Regression

Data 100 Final Review

Suraj Rampure, Allen Shen

Regression vs. Classification

Regression is the problem of creating a model that takes in a point and outputs a real number. We've seen regression in the form of Ordinary Least Squares, Ridge Regression, and LASSO Regression.

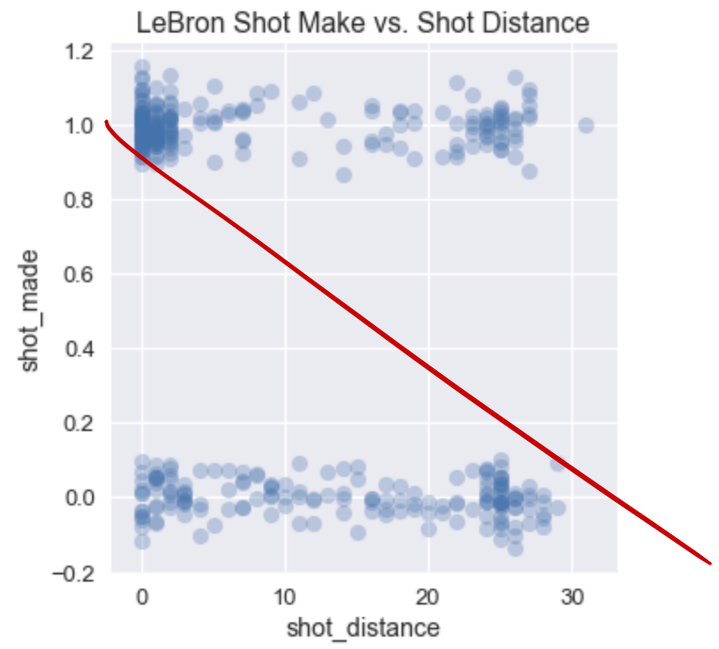
On the other hand, **classification** is the problem of creating a model that takes in a point and outputs a discrete **label**.

A very basic example:

- Regression would allow us to predict a student's final exam grade, given their grades on the midterm and homeworks.
- Classification would allow us to predict whether or not a patient has a disease.

Example (from textbook): Suppose we have LeBron's shot data for a particular season. Specifically, we have the distance from which the shot was taken, and whether or not it went in.

We want to build a model that will allow us to predict whether or not a new shot will go in.



Sure, we *can* fit a standard linear regression model to this, and interpret the output as the *probability that the shot will go in*. For example, we can say if the predicted value is over 0.5, he will make the shot. **However, values aren't restricted to** $[0, 1]$**!**

Probability Recap, Sigmoid Function

A probability density function $f(x)$ is valid iff it satisfies the following conditions:

- $0 \leq f(x) \leq 1$, for all $x \in \mathbb{X}$
- $\sum_{x \in \mathbb{X}} f(x) = 1$

We need some function that maps $\mathbb{R} \rightarrow [0, 1]$. Our choice is $\sigma(x)$:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

This is known as the **sigmoid** function, and it satisfies the following property:

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$$

$$E[y|x] = x^T\beta$$

With standard linear regression, to fit our model $E[Y|x] = x^T\beta$, we found the β that minimizes

$$\min_{\beta} ||y - X\beta||_2^2$$

$$E[y|x] = x^T\beta$$

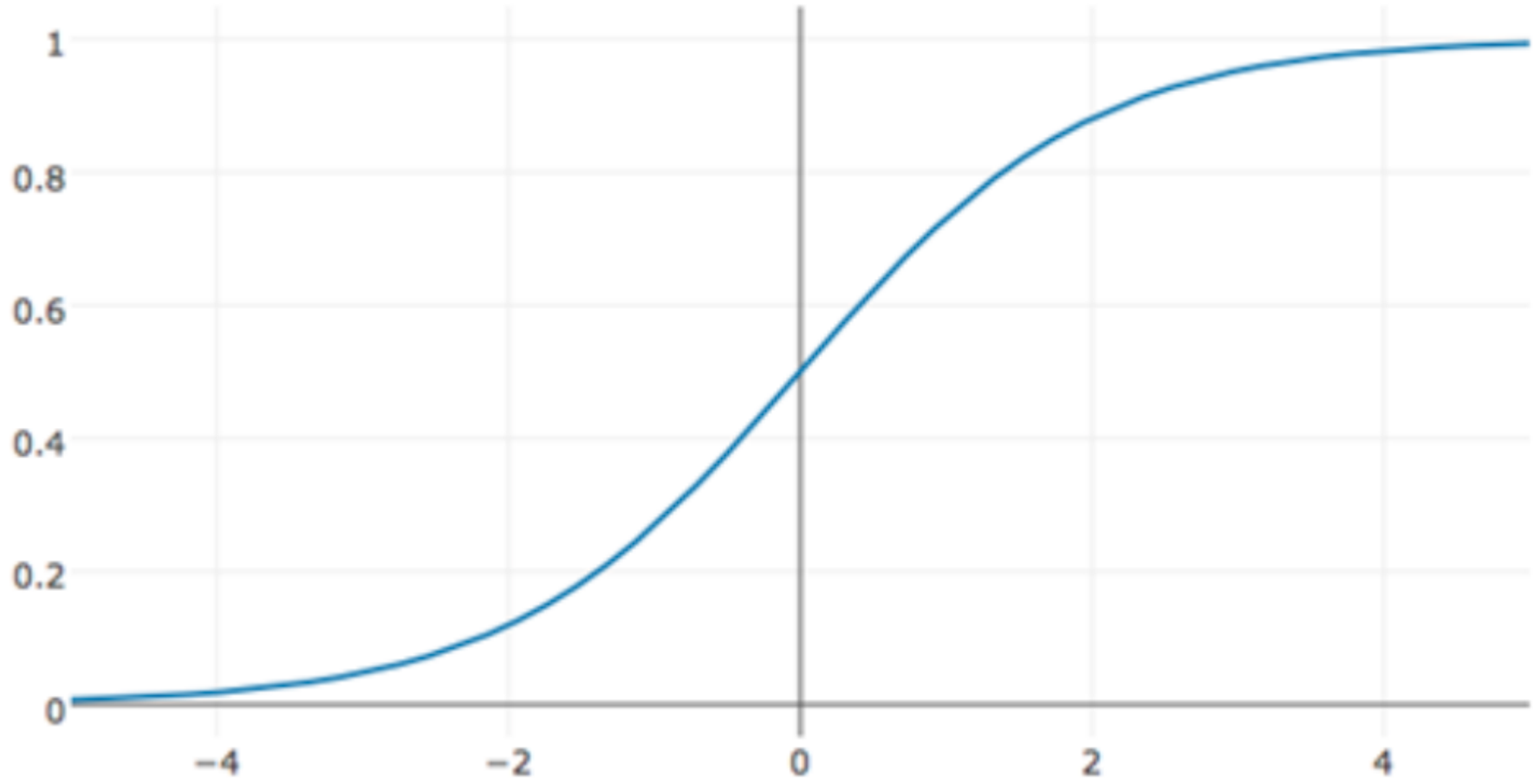
However, with **logistic regression**, we instead model using the following (assuming our classification is binary):

$$\mathbb{P}_\beta(Y = 1|x) = \sigma(x^T\beta) = \frac{1}{1 + \exp(-x^T\beta)}$$

This is the probability that our test point x belongs to class 1 (as opposed to class 0).

Important: The output of logistic regression on its own is not a classification. We need to decide a *cutoff*, or decision boundary, in order to complete our classifier.

Output of $\sigma(x)$



Loss Function for Logistic Regression

The loss function we use for logistic regression is what is known as **cross entropy loss**. It has information-theoretic foundations, and is preferred over L2 loss for a number of reasons.

Average cross entropy loss is of the form

$$L(\beta) = -\frac{1}{n}\sum_{i=1}^{n}(y_i x_i^T \beta + \log(\sigma(-x_i^T \beta)))$$

This cannot be determined analytically, unlike the solution to OLS. We must use a numerical method, such as gradient descent, to determine β^*.

Linear Separability

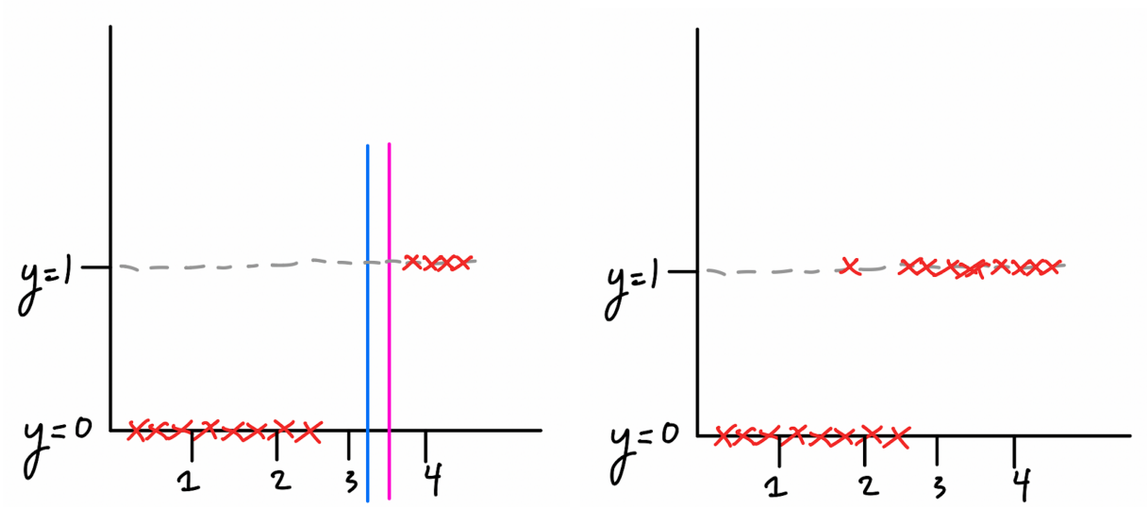
The goal of ~~linear~~ logistic regression is to model the probability of a point belonging to a class. We only do this when there is some level of uncertainty, i.e. overlap, in our training set.

In some cases, we are able to draw a *linear decision boundary* to separate our data.

Above, we see that our data is **linearly separable**. We can draw a line (infinitely many lines, in fact) between the clusters of red dots and green dots.

This is not the case in the second example.

Again, let's look at data in 1D, but plotted in 2D (one dimension is the value of our variable, the second dimension is the label, 0 or 1 --- this is equivalent to the drawings on the previous slide).

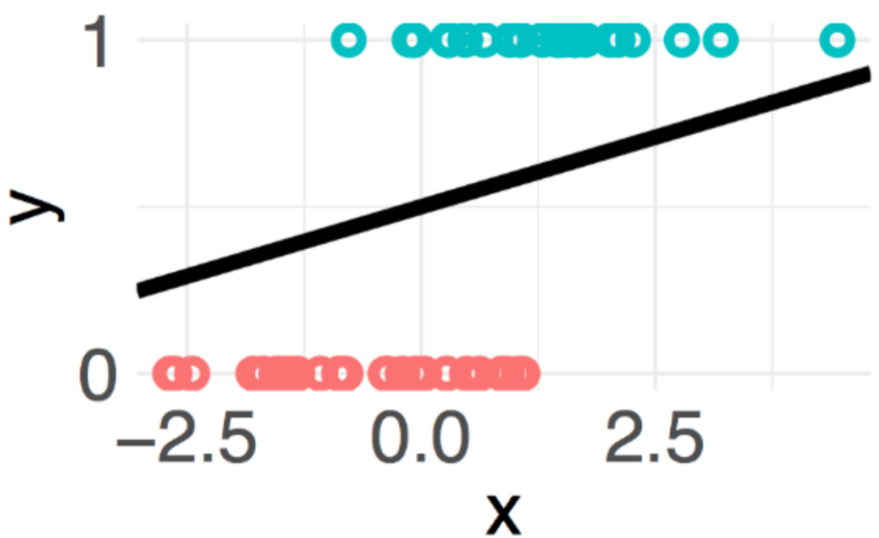


On the left, we have an example of linearly separable data. In the blue and purple are two possible hyperplanes that can separate our data. There would be no use in using logistic regression here.

However, on the right, we have non-linearly separable data. In this case we would use a tool like logistic regression to model probabilities.

IMPORTANT: Linear Separability Depends on the Dimension of the Data!

A set of d-dimensional points is linearly separable iff we can draw a degree $d - 1$ hyperplane to separate the points.



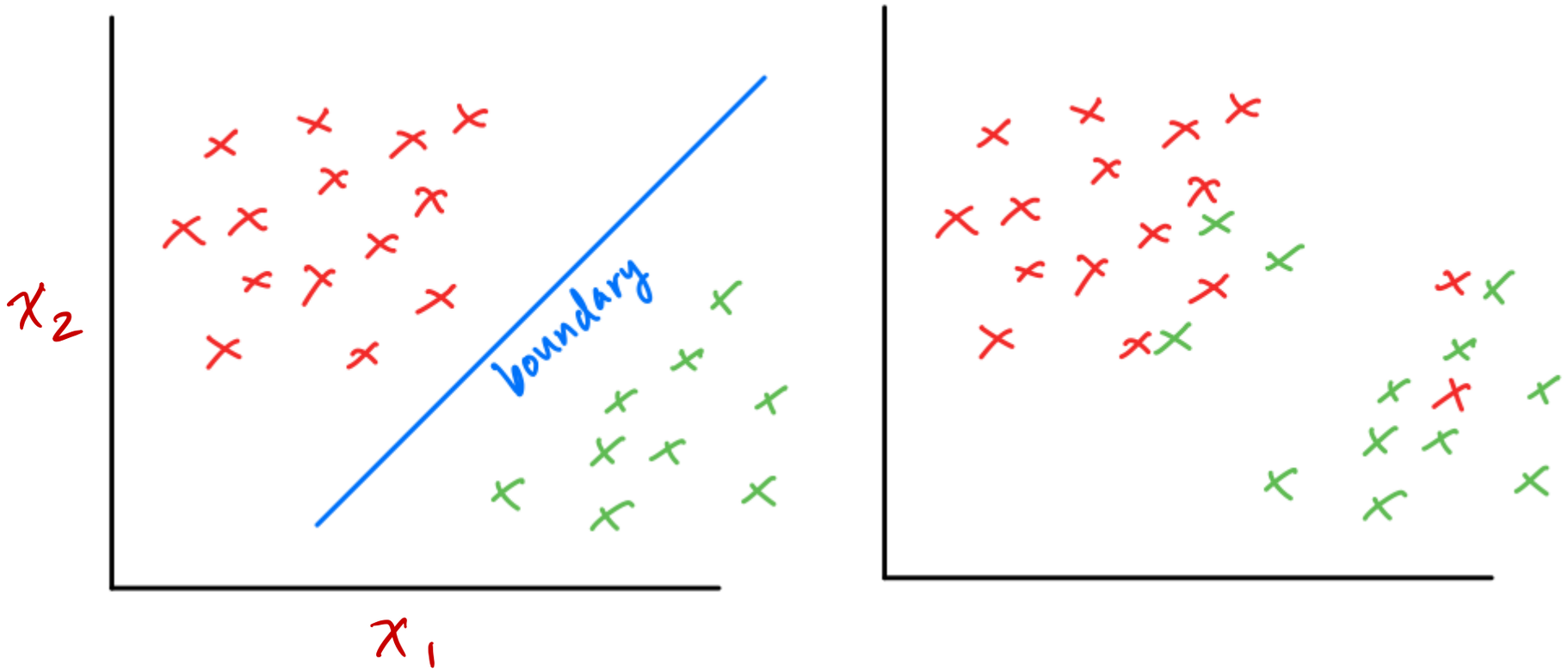
data : $d = 1$

hyperplane $d = 0$

$x = a$

This data **is not** linearly separable. This problem (from Disc 8) is intentionally misleading; the points are in 1D, however the class labels are represented in two ways (y axis and color). We cannot draw a degree 0 line (i.e. something of the form $x = a$) to separate this data.

Linear Separability in Two Dimensions



Here, we have examples of both linearly separable and non-linearly separable data in two dimensions. Here, our data is truly two dimensional, as our feature space has two components --- an x_1 and a x_2. The class is represented by the color (Y).

Evaluating the Effectiveness of a Classifier

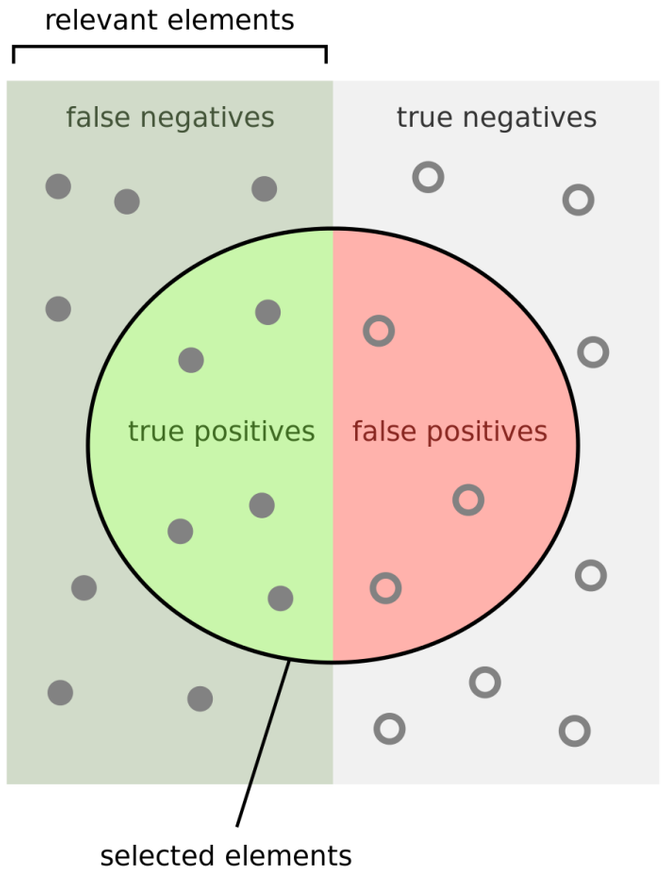
Suppose we train a binary classifer, and suppose `y` represents actual values, and `y_pred` represents predicted values.

Recall (no pun intended) the following definitions:

- True Positives: `TP = np.count_nonzero((y == y_pred) & (y_pred == 1))`
- True Negatives: `TN = np.count_nonzero((y == y_pred) & (y_pred == 0))`
- False Positives: `FP = np.count_nonzero((y != y_pred) & (y_pred == 1))`
- False Negatives: `FN = np.count_nonzero((y != y_pred) & (y_pred == 0))`

Then, we have the following definitions:

- Precision = $\frac{TP}{TP+FP}$, measures the number of predicted true values that are actually true
- Recall = $\frac{TP}{TP+FN}$, measures the number of actually true values that are marked true ("detection rate")



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Example

Suppose you create a classifier to predict whether or not an image contains a picture of a goat. You test it on 23 images.

- There were 12 true images of goats. Your classifier predicted 9 of them to be goats, and 3 to not be a goat.
- There were 11 images that did not contain goats. Your classifier predicted 3 of them to be goats, and the remaining 8 to not be goats.

Determine the precision and recall of your goat classifier.

$TP = 9$

$FN = 3$

$FP = 3$

$TN = 8$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{9}{9+3} = \frac{3}{4}$$

↑ coincidence

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{9}{9+3} = \frac{3}{4}$$

Practice: Classification (T/F)

True/False: In logistic regression, predictor variables (x) are continuous, with values in the range $[0, 1]$.

True/False: In two-class binary classification, the output is continuous, with values in the range $[0, 1]$.

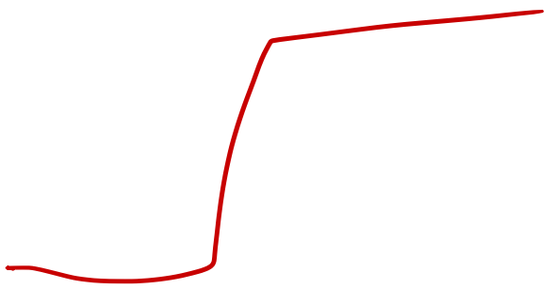
True/False: In logistic regression, we calculate the weights β^* as $\beta^* = (X^T X)^{-1} X^T y$ and then fit responses as $y = \sigma(x^T \beta)$.

Practice: Logistic Regression (T/F)

True/False: If no regularization is used and the training data is linearly separable, the parameters will tend towards positive or negative infinity.

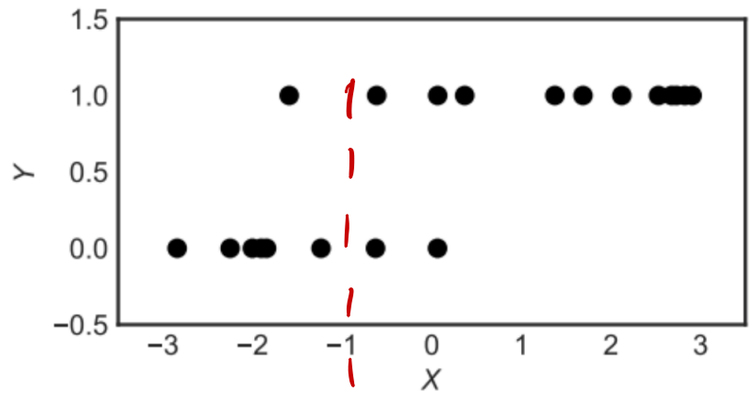
True/False: L_1 reguarlization can help us select a subset of the features that are important.

True/False: After regularization, we expect the training accuracy to increase and the test accuracy to decrease.



Practice: Interpreting Logistic Regression

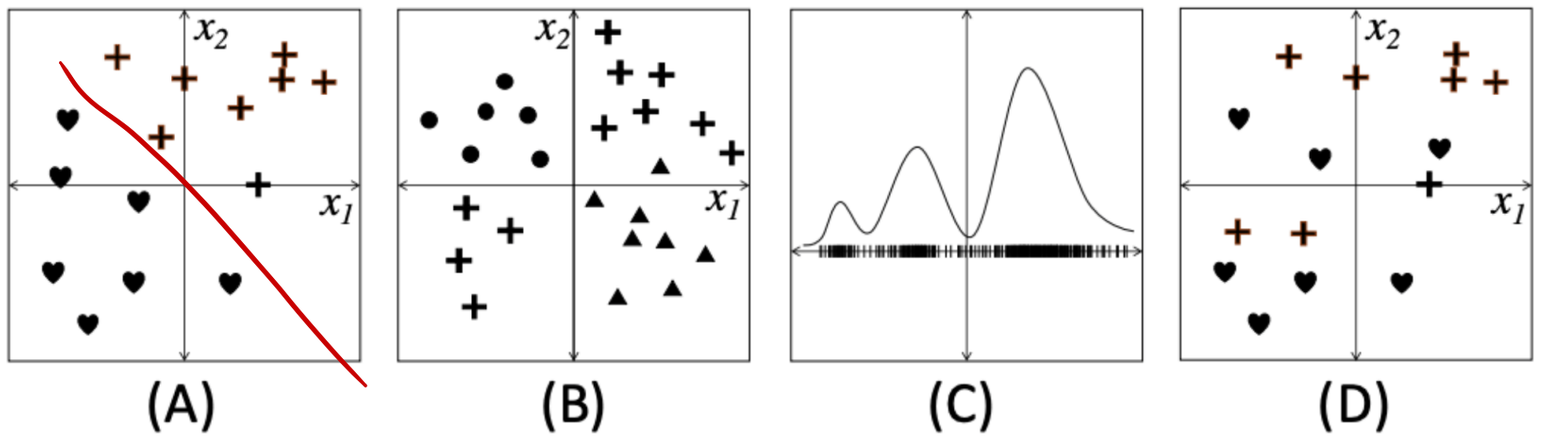
[2 Pts] Suppose you are given the following dataset $\{(x_i, y_i)\}_{i=1}^{n}$ consisting of x and y pairs where the covariate $x_i \in \mathbb{R}$ and the response $y_i \in \{0, 1\}$.



Given this data, the value $\mathbf{P}\,(Y = 1 \,|\, x = -1)$ is likely closest to:

○ 0.95 ● 0.50 ○ 0.05 ○ -0.95

Practice: Separability



(1) Which of the above plots represents a **linearly separable binary classification** task?

◉ (A) ○ (B) ○ (C) ○ (D)

(2) Which of the above plots represents a **binary classification** task that is **not linearly separable**?

○ (A) ○ (B) ○ (C) ◉ (D)

(3) Which of the above plots represents a **multi-class classification task**?

○ (A) ◉ (B) ○ (C) ○ (D)