Logistic Regression

Kanokkorn Pimcharoen

Brain Code Camp

Speaker

Kanokkorn Pimcharoen

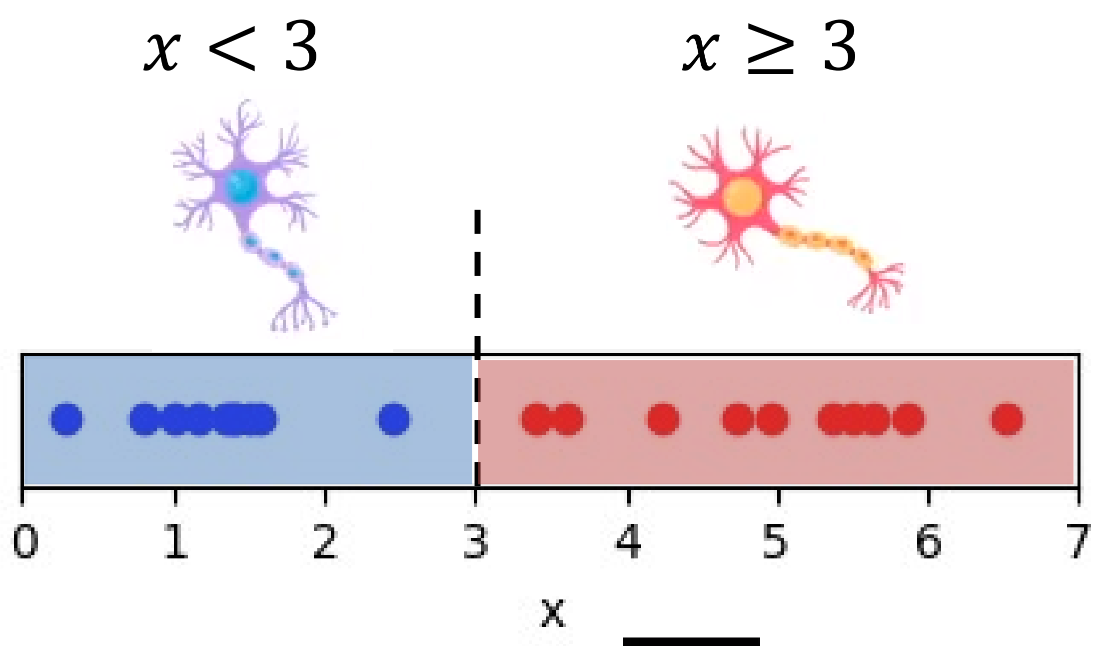
Researcher, Computational Physics and AI

Nanoinformatics and Artificial Intelligence

Research Team, NANOTEC, NSTDA

Hobbies: Travel and Coffee

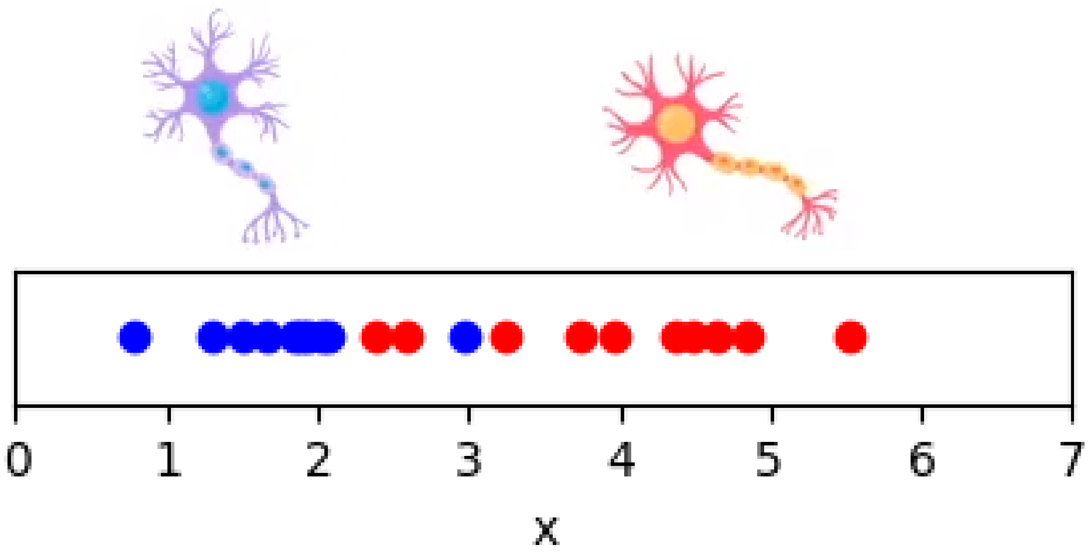
Classification Problem



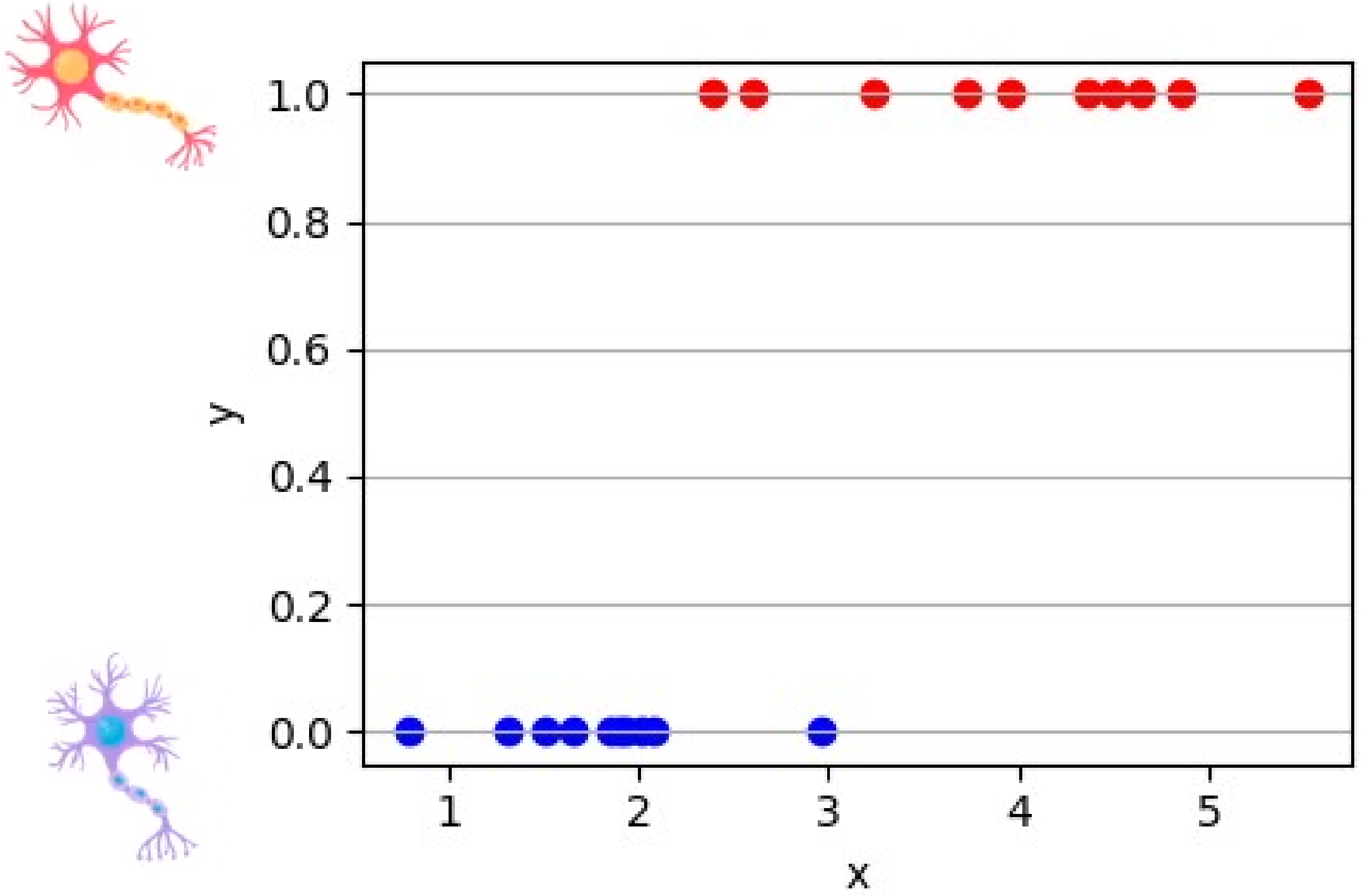
Spike counts

Classification Problem

Class 0 (y=0) Class 1 (y=1)



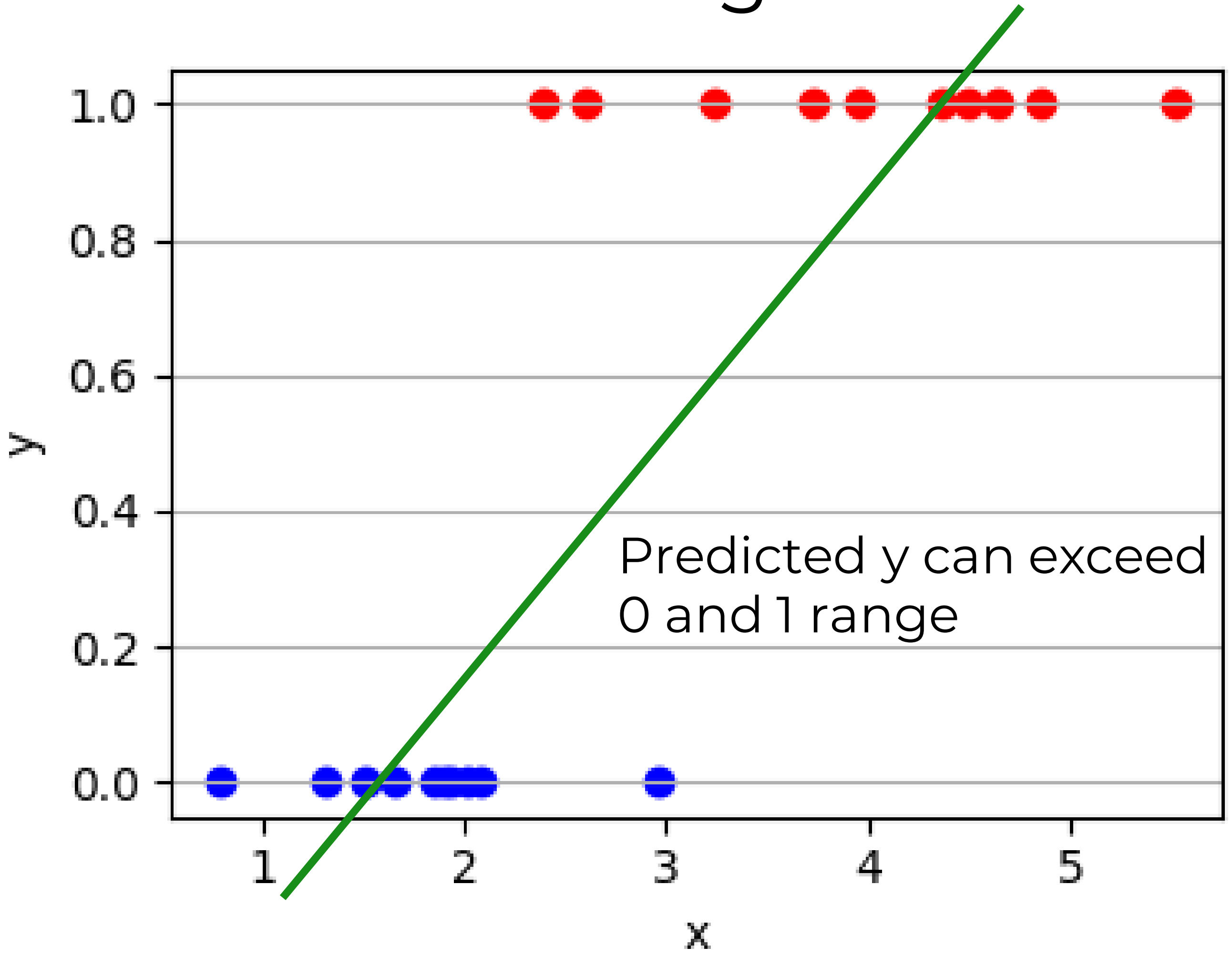
Spike counts



Spike counts

Generalized Linear Models

Linear Regression

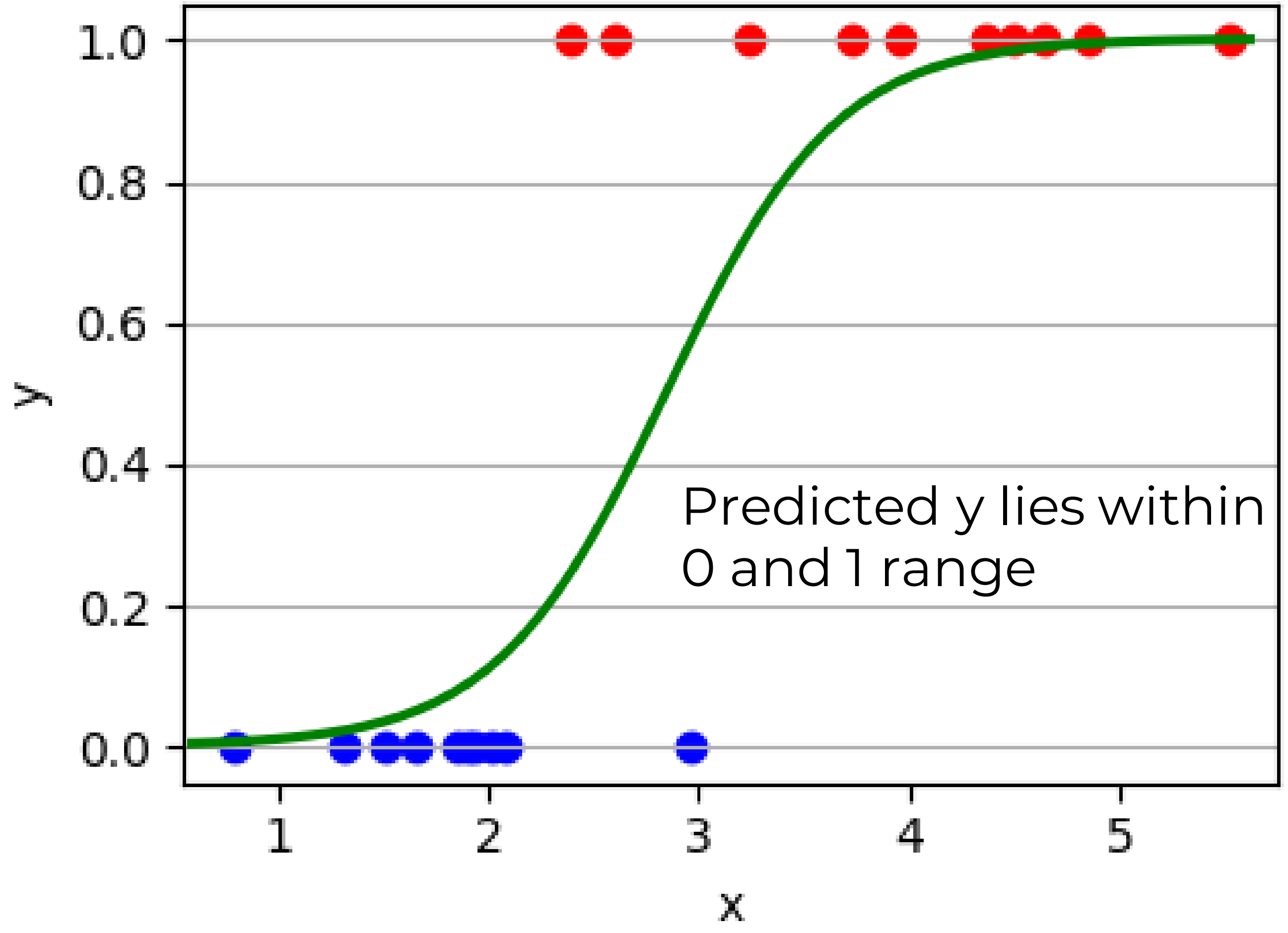


$$y = w_0 + w_1\, x$$

w_1: slope

w_0: y-intercept

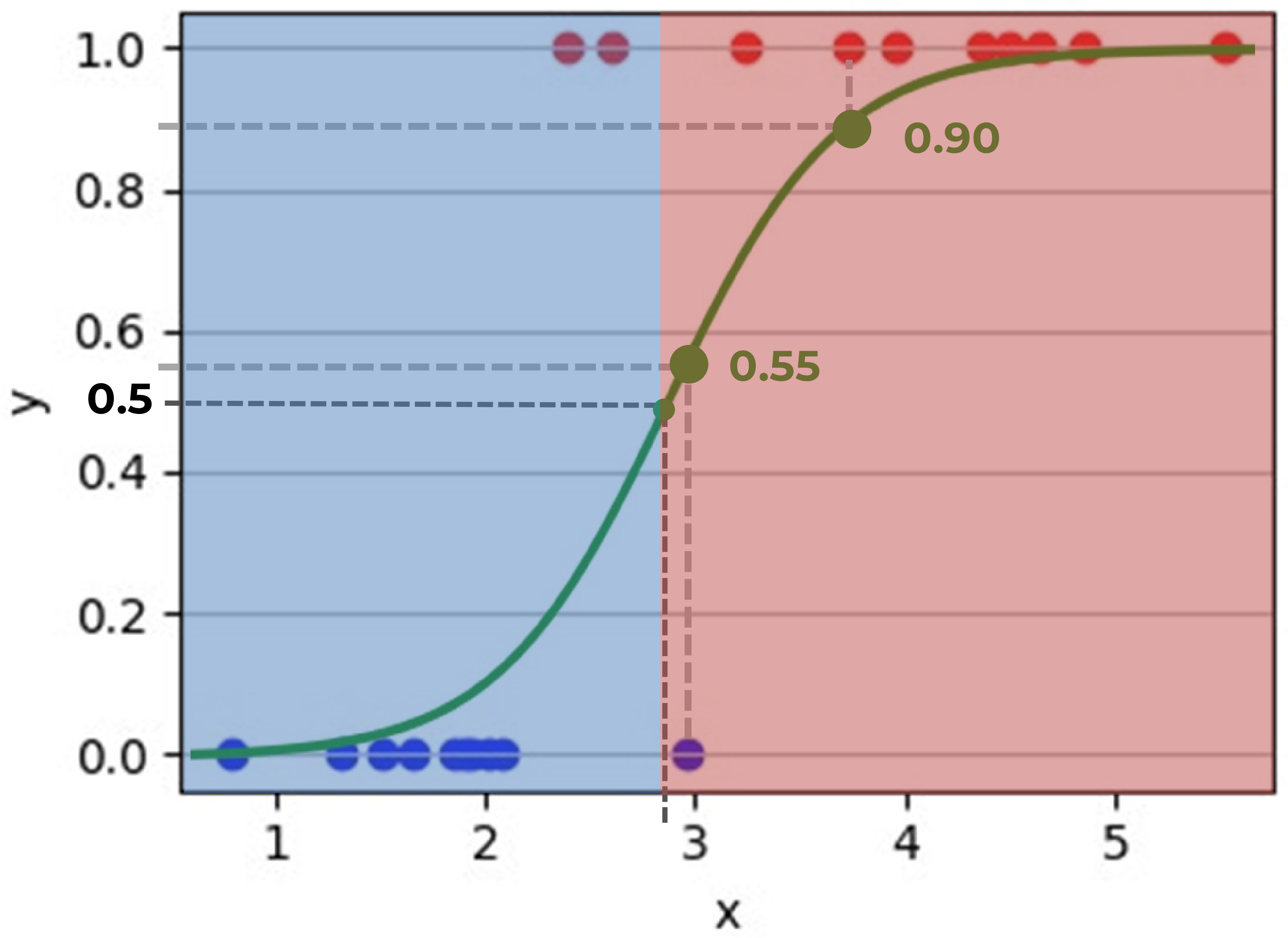
Logistic Regression



$$y = \frac{1}{1 + e^{-(w_0 + w_1\, x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1\, x$$

$Probability$ $\qquad$ $log - odds$

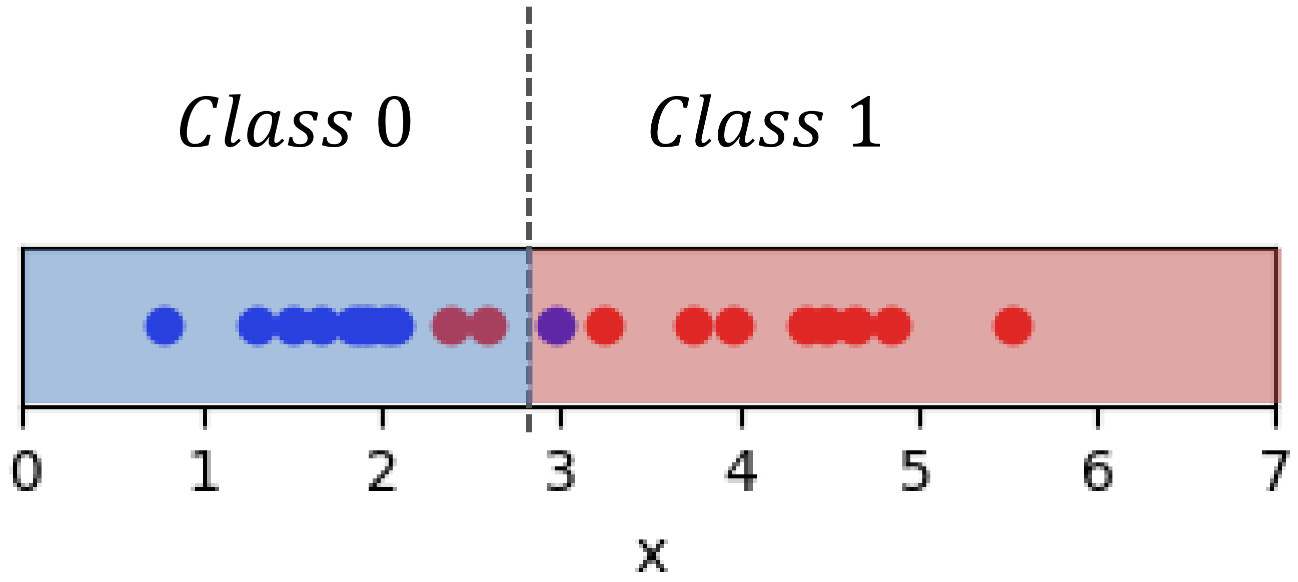
Logistic Regression



$y = Probability\ of\ class\ 1$

$y \geq 0.5 \ \rightarrow Class\ 1 \leftarrow x \geq 2.8$

$y < 0.5 \ \rightarrow Class\ 0 \leftarrow x < 2.8$

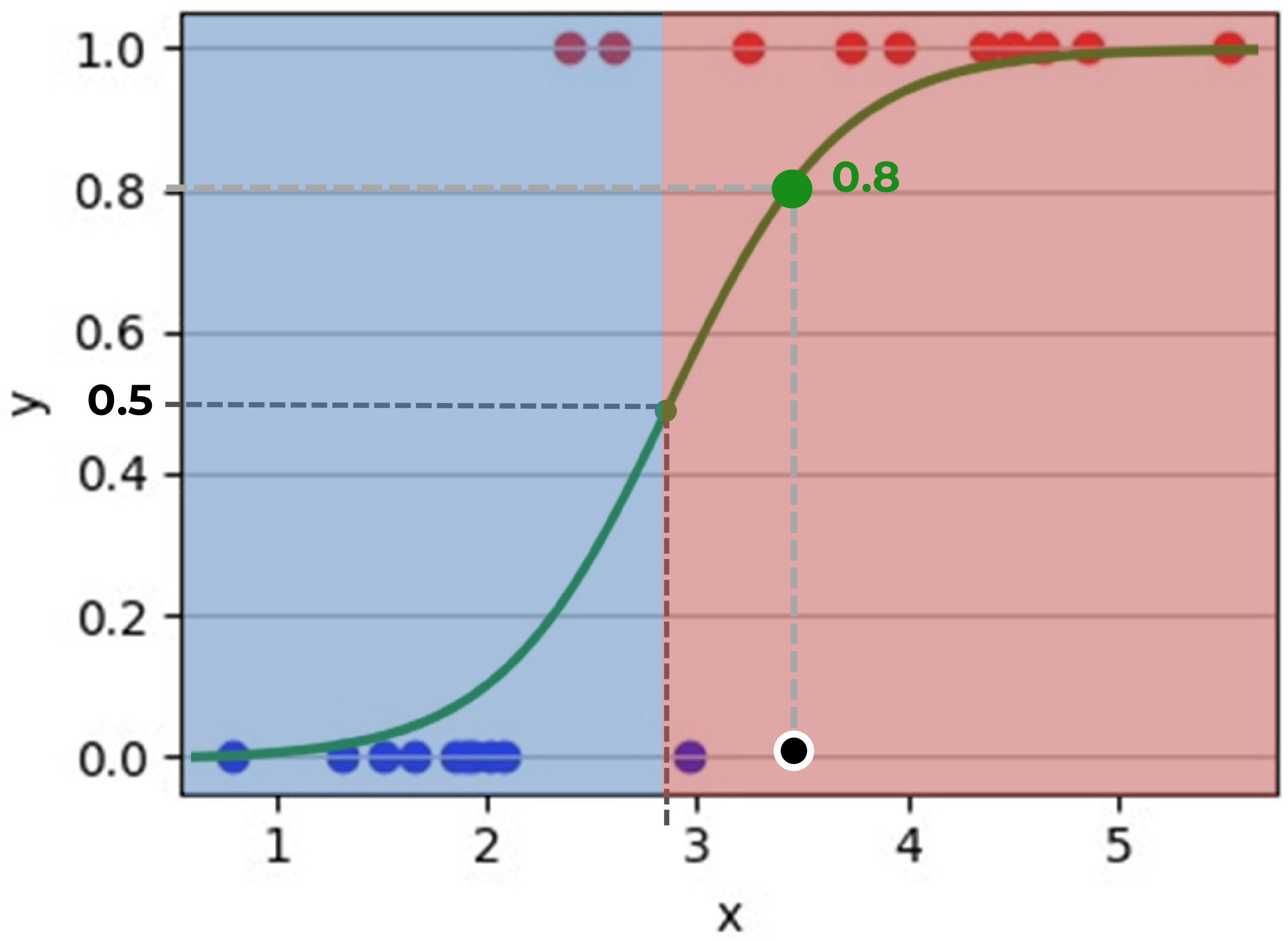


$$y = \frac{1}{1 + e^{-(w_0 + w_1 x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1 x$$

Decision Boundary

Brain Code Camp

Logistic Regression

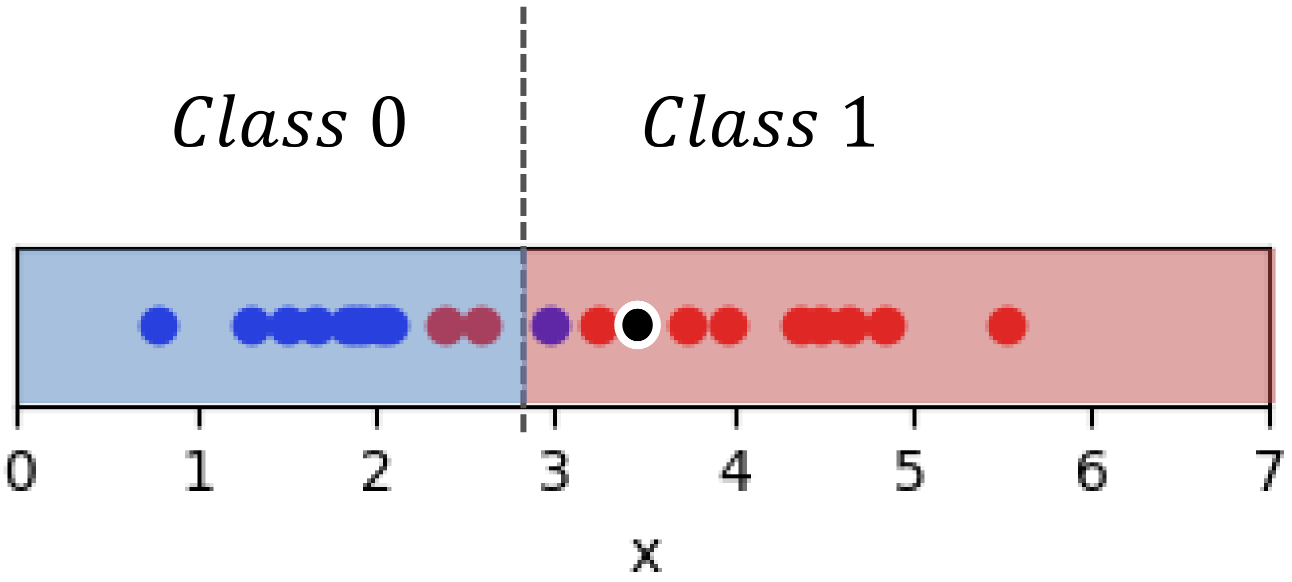


$$y = \frac{1}{1 + e^{-(w_0 + w_1 x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1 x$$

$y = Probability\ of\ class\ 1$

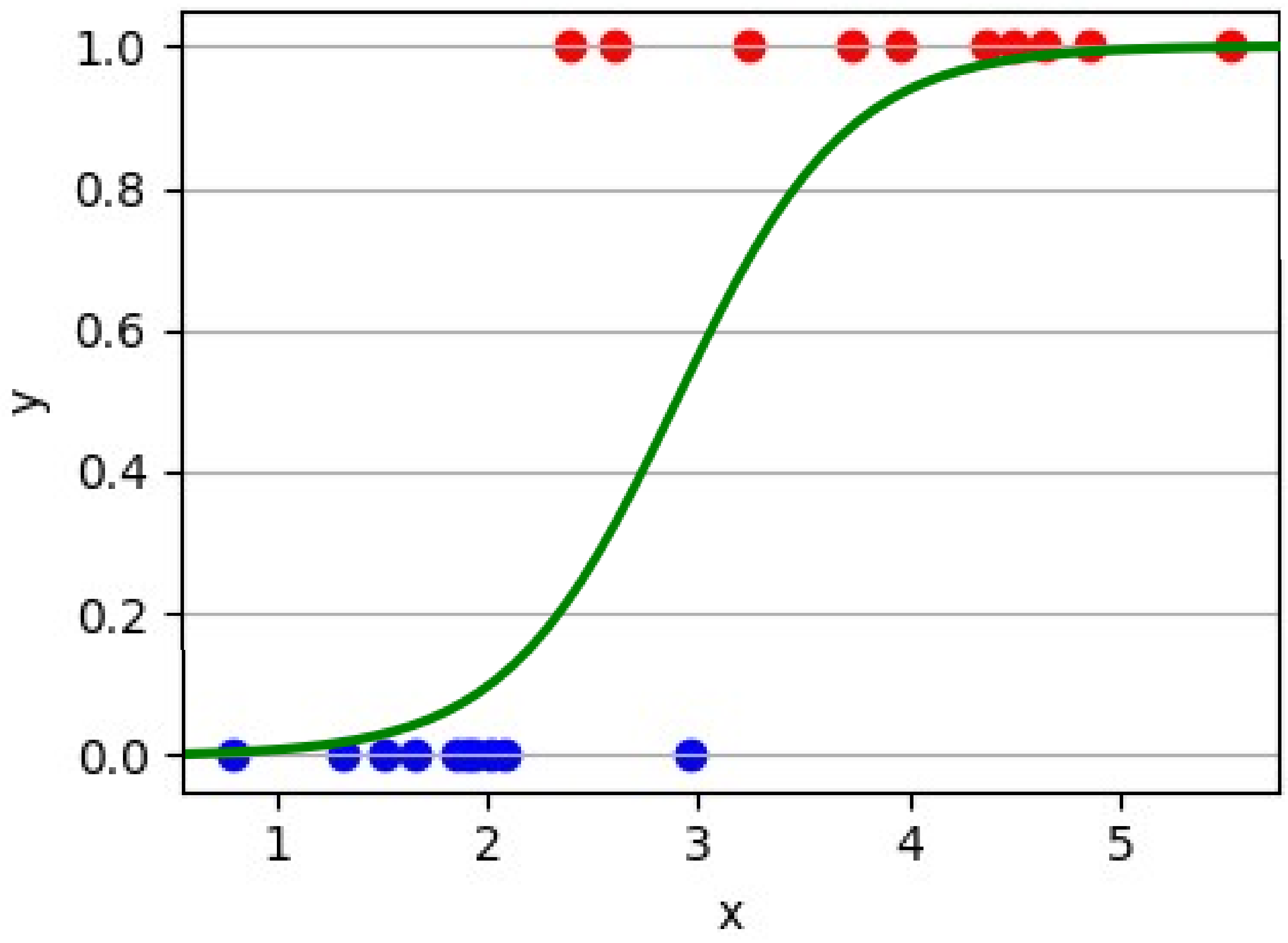
$y \geq 0.5 \ \rightarrow Class\ 1 \leftarrow x \geq 2.8$

$y < 0.5 \ \rightarrow Class\ 0 \leftarrow x < 2.8$



Decision Boundary

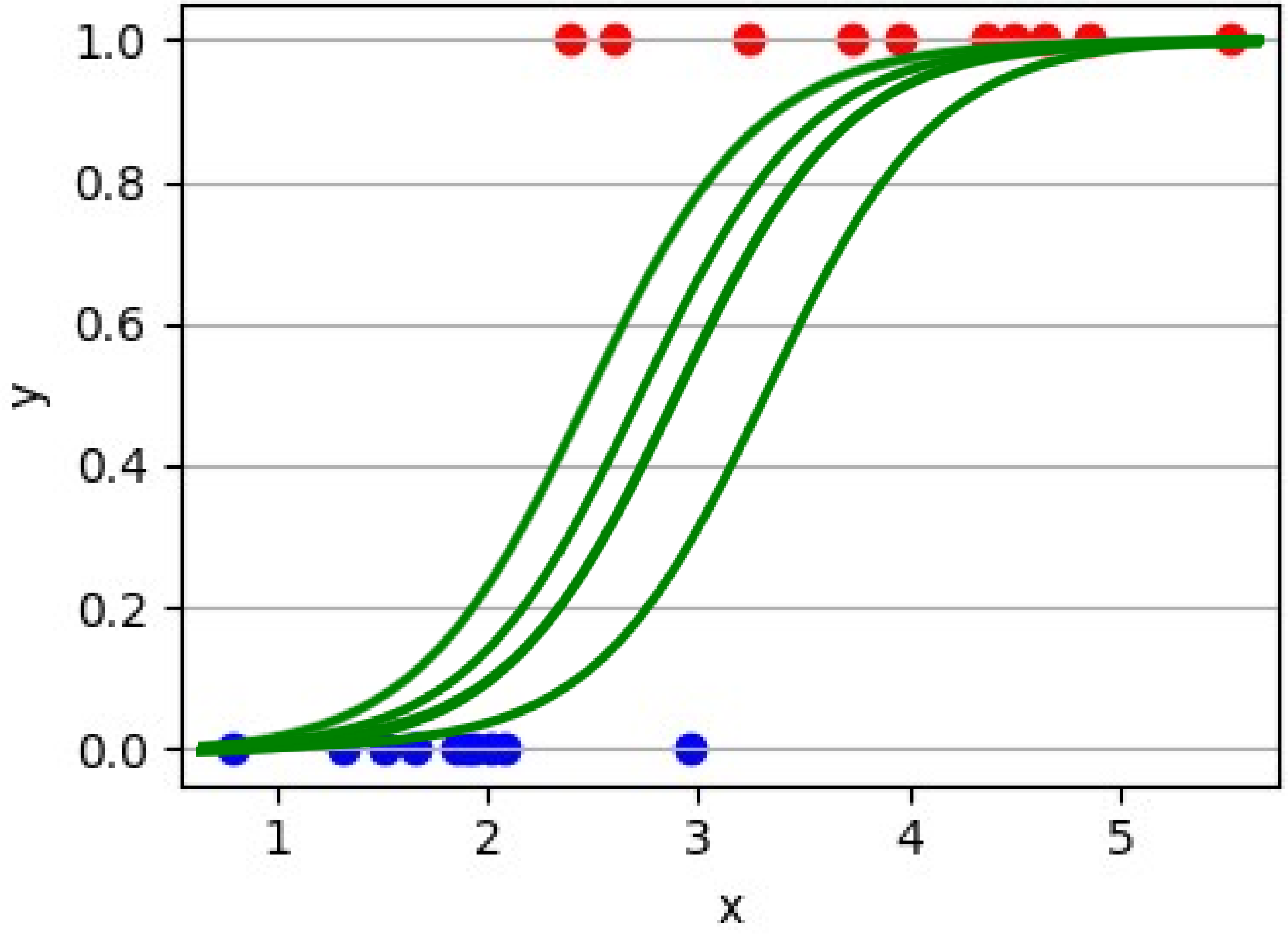
Logistic Regression: Optional



$$y = \frac{1}{1 + e^{-(w_0 + w_1\, x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1\, x$$

Brain Code Camp

Logistic Regression: Optional



$$y = \frac{1}{1 + e^{-(w_0 + w_1 x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1 x$$

Maximum Likelihood Estimation

$$\mathcal{L}(x) = \prod_i \left[\frac{1}{1 + e^{-(w_0 + w_1 x_i)}}\right] \times \prod_j \left[1 - \frac{1}{1 + e^{-(w_0 + w_1 x_j)}}\right]$$

To obtain the values of w_0 and w_1 that maximize $\mathcal{L}(x)$, an iterative solver is applied.

When the difference of $\mathcal{L}(x)$ between two consecutive iterations becomes lower than tolerance, the solver is converged and yielded the optimal parameters w_0 and w_1.

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.**LogisticRegression**(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

Parameters:

solver : *{'lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga'}, default='lbfgs'*

'lbfgs' :
- Small-to-medium dataset
- Support L2 regularization only

'liblinear' :
- Small dataset
- Support both L1 and L2 regularizations

'newton-cg' :
- Large dataset
- The computation time grows with increasing number of features
- Support L2 regularization only

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

Tolerance for stopping criteria.

max_iter : *int, default=100*

Maximum number of iterations taken for the solvers to converge.

sklearn.linear_model.LogisticRegression

class sklearn.linear_model.**LogisticRegression**(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

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: The choice of the algorithm depends on the penalty chosen. Supported penalties by solver:

- 'lbfgs' - ['l2', None]
- 'liblinear' - ['l1', 'l2']
- 'newton-cg' - ['l2', None]
- 'newton-cholesky' - ['l2', None]
- 'sag' - ['l2', None]
- 'saga' - ['elasticnet', 'l1', 'l2', None]

C : *float, default=1.0*

Inverse of regularization strength; must be a positive float.

$$\mathcal{L}(x,y) + \lambda R(w) \longrightarrow C\,\mathcal{L}(x,y) + R(w)$$

Setting C to high value will reduce the effect of regularization.

sklearn.linear_model.LogisticRegression

class `sklearn.linear_model.LogisticRegression`(*penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*) [source]

$$y = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m)}}$$

Attributes:

classes_ : *ndarray of shape (n_classes,)*

A list of class labels known to the classifier.

coef_ : *ndarray of shape (1, n_features) or (n_classes, n_features)*

Coefficient of the features in the decision function.

`coef_` is of shape (1, n_features) when the given problem is binary. In particular, when `multi_class='multinomial'`, `coef_` corresponds to outcome 1 (True) and `-coef_` corresponds to outcome 0 (False).

$w_1, w_2, \ldots, w_m$

intercept_ : *ndarray of shape (1,) or (n_classes,)*

Intercept (a.k.a. bias) added to the decision function.

If `fit_intercept` is set to False, the intercept is set to zero. `intercept_` is of shape (1,) when the given problem is binary. In particular, when `multi_class='multinomial'`, `intercept_` corresponds to outcome 1 (True) and `-intercept_` corresponds to outcome 0 (False).

w_0

Logistic Regression

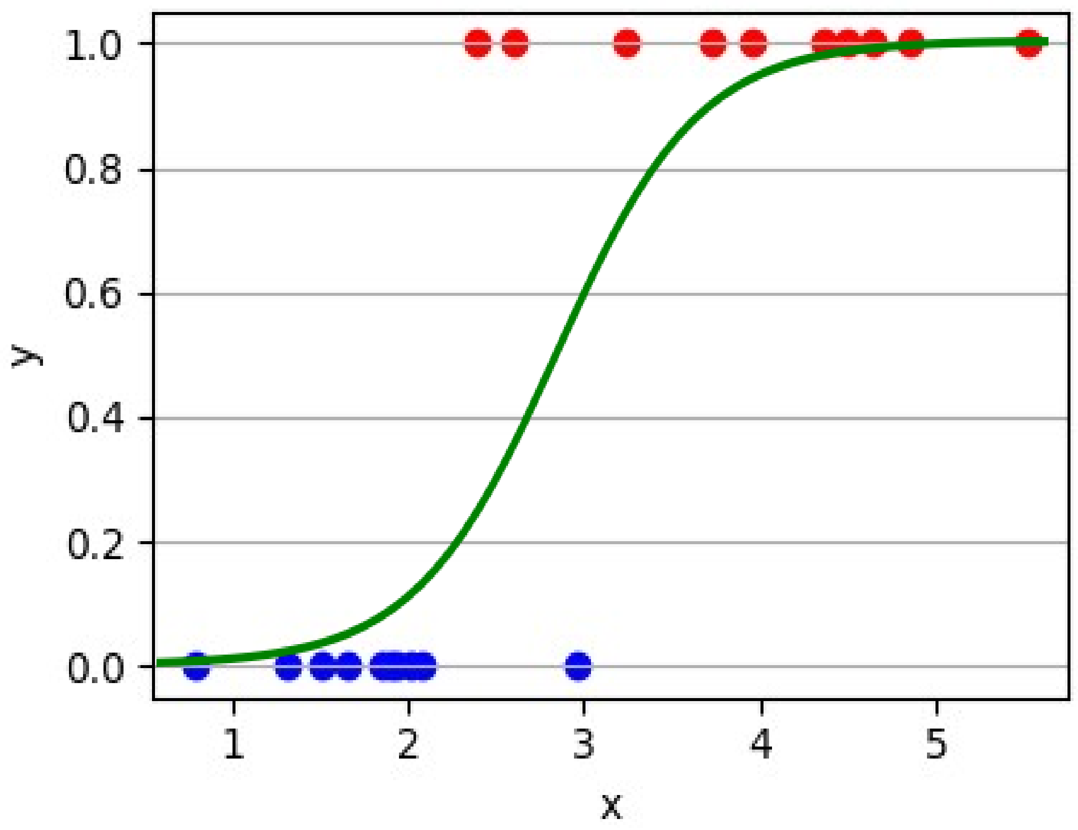
Dataset

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

shape = $(n, 1)$ shape = $(n, 1)$

Dataset for binary classification (n=20)



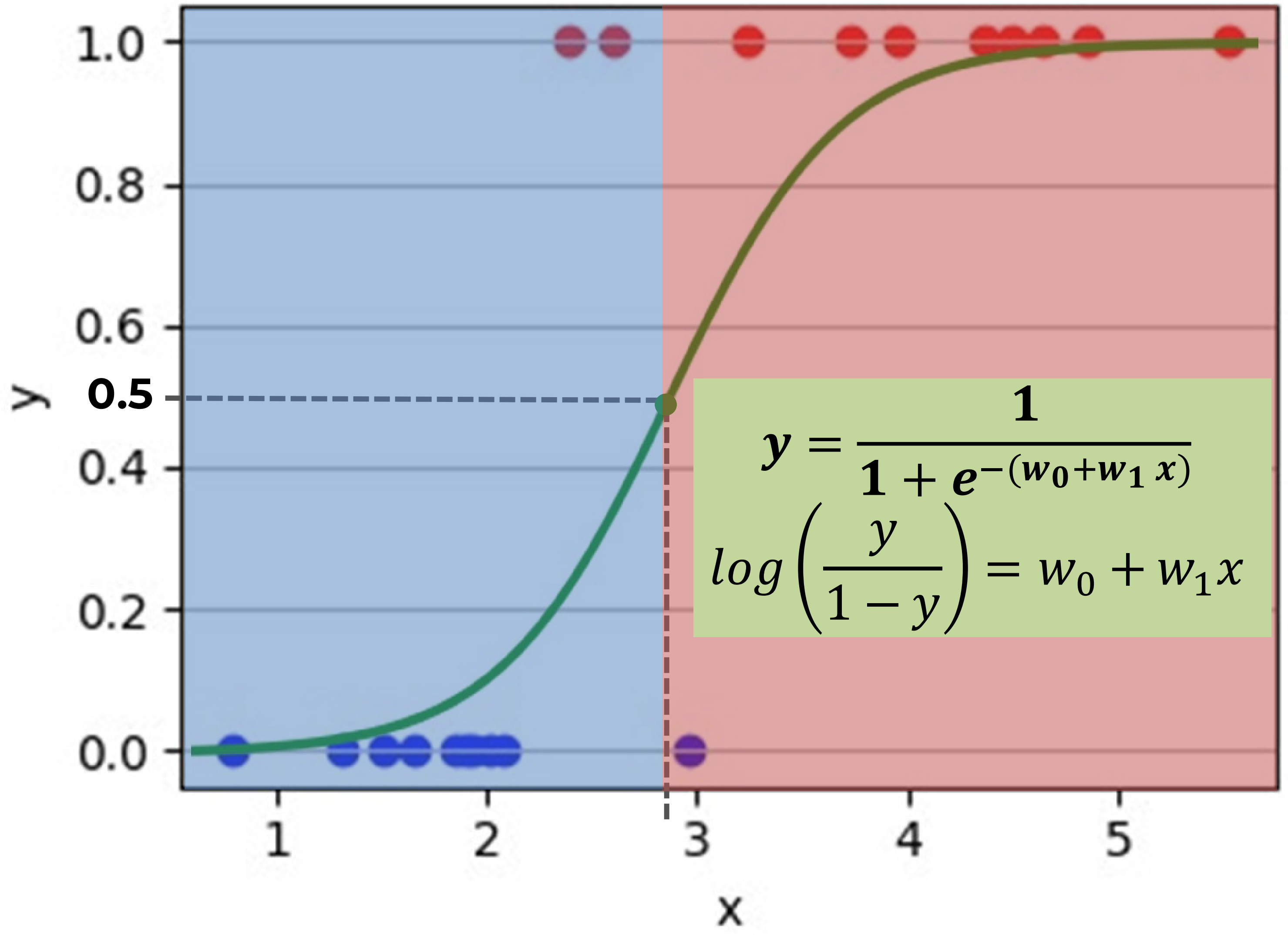

```
# Import a necessary modules
from sklearn.linear_model import LogisticRegression

# Create the model
clf = LogisticRegression()

# Train the model
clf.fit(x,y)

# Obtain model parameters
w0 = clf.intercept_
w1 = clf.coef_

# Make prediction
y_pred = clf.predict(x_test)
```

$$y = \frac{1}{1 + e^{-(w_0 + w_1 x)}}$$

Logistic Regression

Kanokkorn Pimcharoen

Logistic Regression: Optional



$$y = \frac{1}{1+e^{-(w_0+w_1\,x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1\,x$$

$y = Probability\ of\ class\ 1$

$y \geq 0.5 \ \rightarrow Class\ 1 \ \leftrightarrow \ w_0 + w_1\,x \geq 0$

$y < 0.5 \ \rightarrow Class\ 0 \ \leftrightarrow \ w_0 + w_1\,x < 0$

$$y = \frac{1}{1+e^{-(w_0+w_1\,x)}} \geq \frac{1}{2}$$

$$2 \geq 1 + e^{-(w_0+w_1\,x)}$$

$$1 \geq e^{-(w_0+w_1\,x)}$$

$$\cancel{\ln 1}^{\,0} \geq -(w_0 + w_1\,x)$$

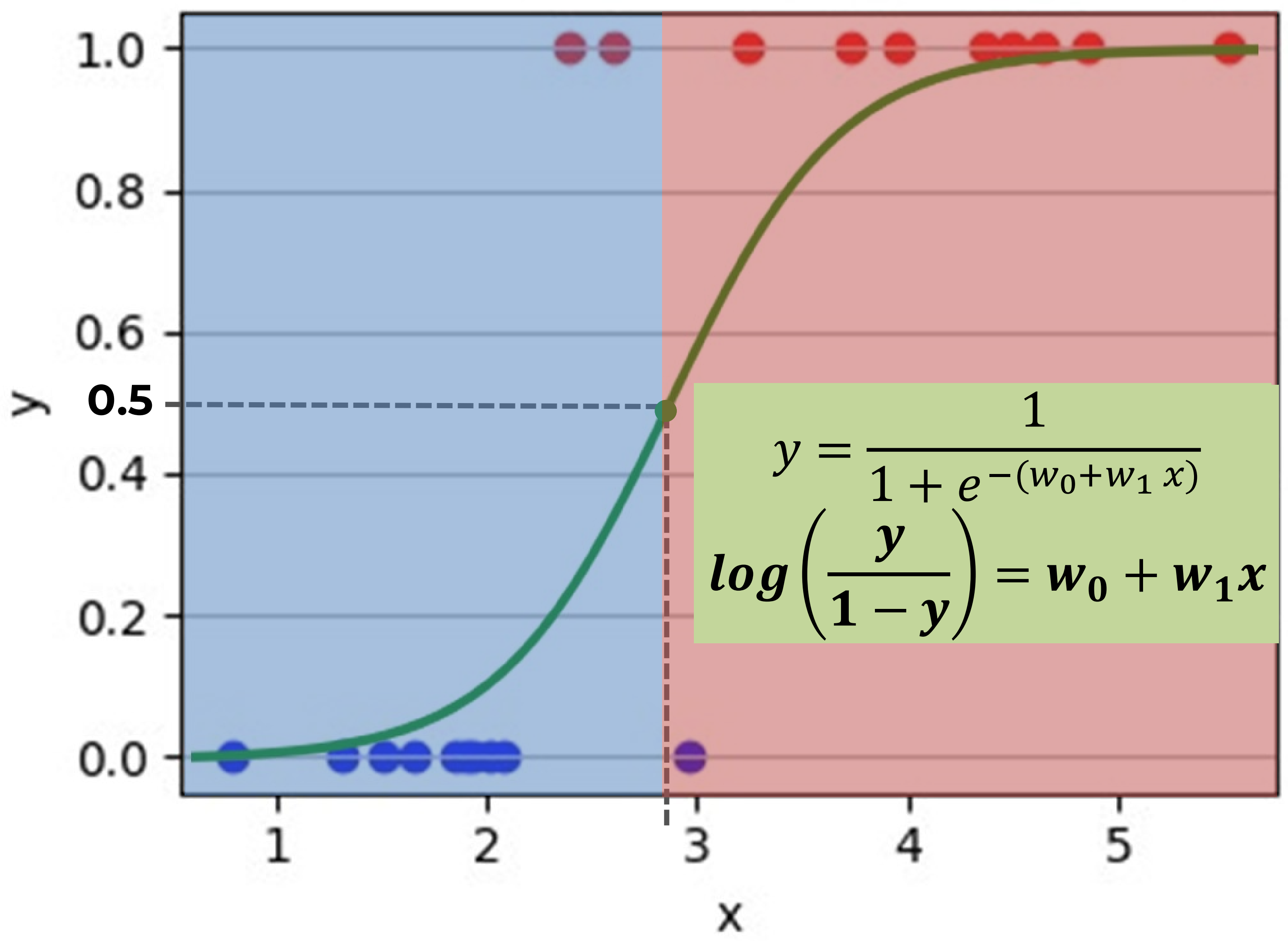
$$w_0 + w_1\,x \geq 0$$

Given $y = 0.5$, decision boundary (DB)?

$w_0 + w_1 x_{DB} = 0$ --> $x_{DB} = -w_0/w_1$

Brain Code Camp

Logistic Regression: Optional



$$y = \frac{1}{1+e^{-(w_0+w_1\,x)}} \quad \text{or} \quad \log\left(\frac{y}{1-y}\right) = w_0 + w_1\,x$$

w_1 : slope w_0 : log-odd-intercept

Probability of class $1 \rightarrow 0$

Probability of class $1 \rightarrow 1$

$$\lim_{y\to 0} \log\left(\frac{y}{1-y}\right) = -\infty \quad \text{and} \quad \lim_{y\to 1} \log\left(\frac{y}{1-y}\right) = +\infty$$

Log-odd has a range of $-\infty$ to $+\infty$

Given $y = 0.5$, decision boundary (DB)?

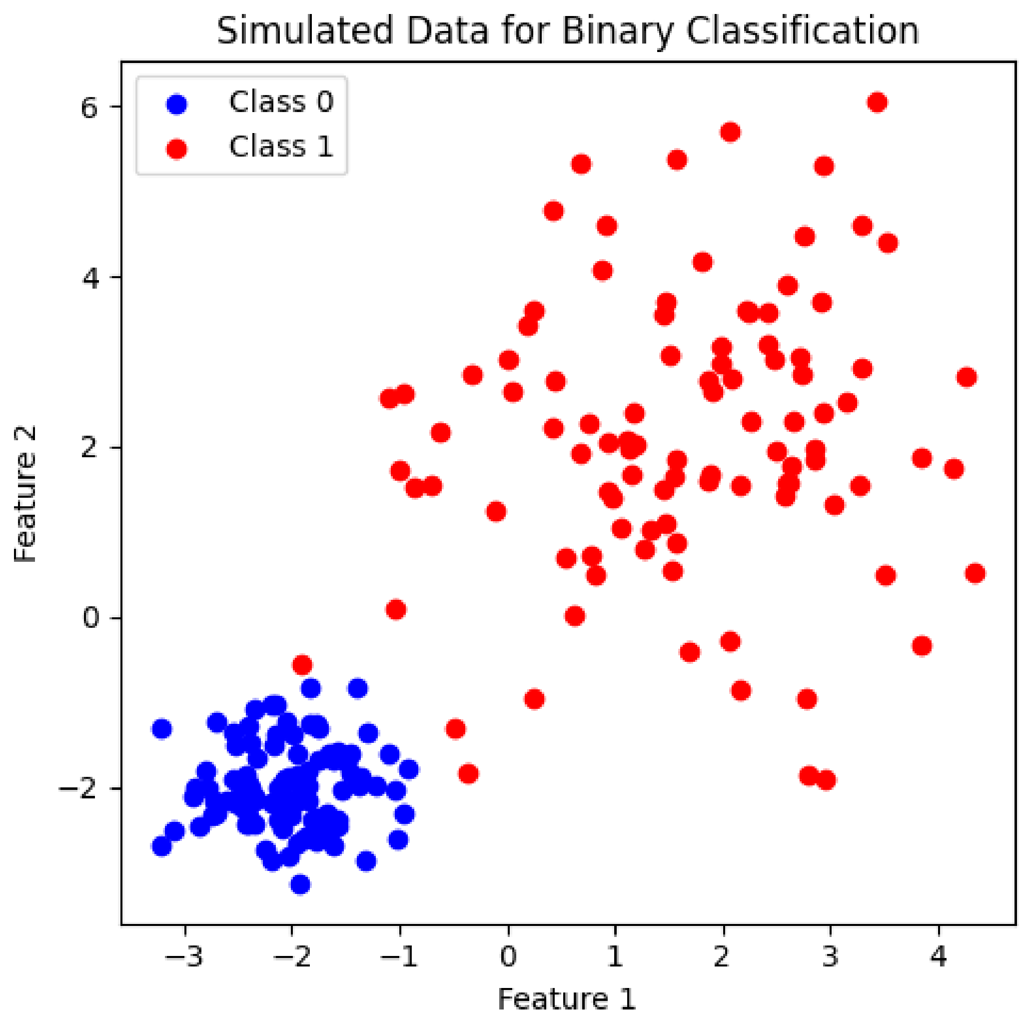
$$\log\left(\frac{0.5}{1-0.5}\right) = \cancelto{0}{\log 1} = w_0 + w_1\,x_{DB} \longleftrightarrow \quad x_{DB} = -w_0/w_1$$

2 Features: $0 = w_0 + w_1\,x_{1,DB} + w_2\,x_{2,DB}$

$$x_{2,DB} = \frac{w_0}{-w_2} + \frac{w_1}{-w_2}\,x_{1,DB}$$

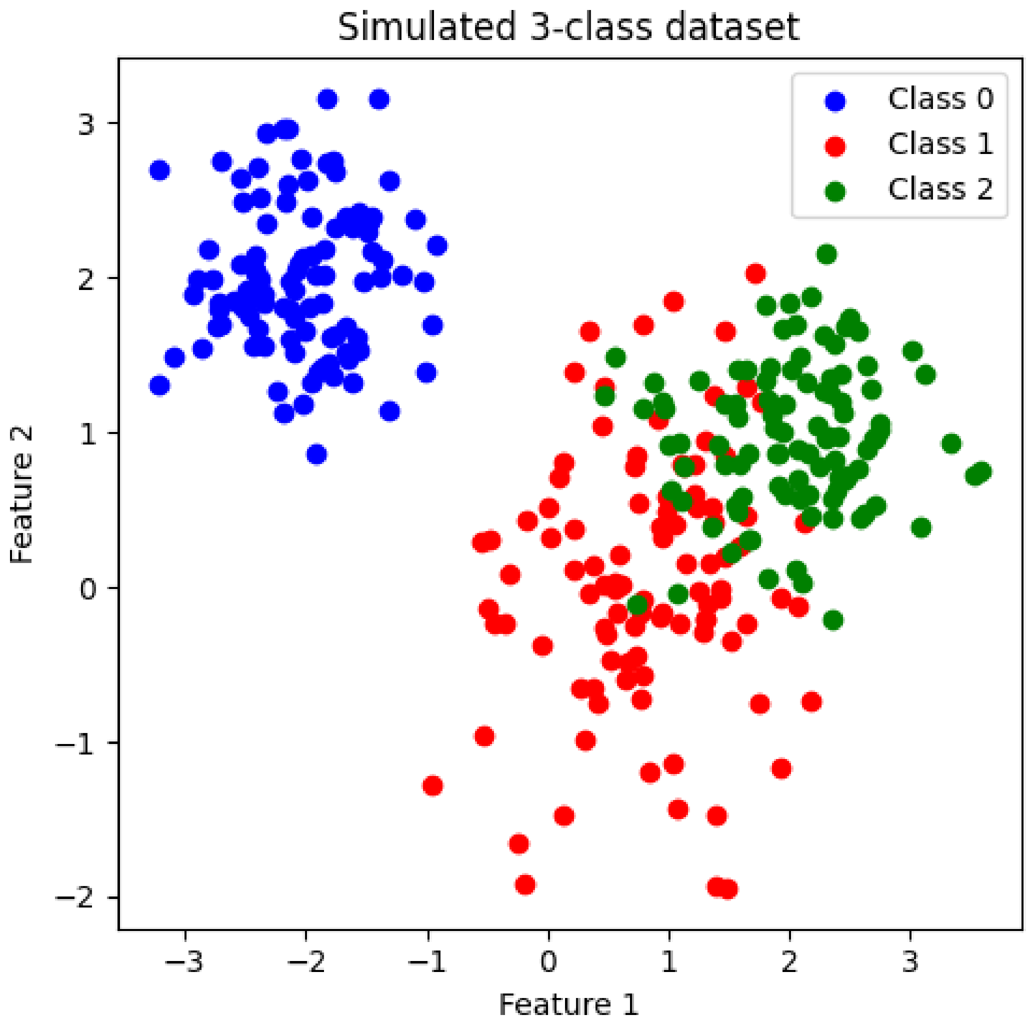
Brain Code Camp

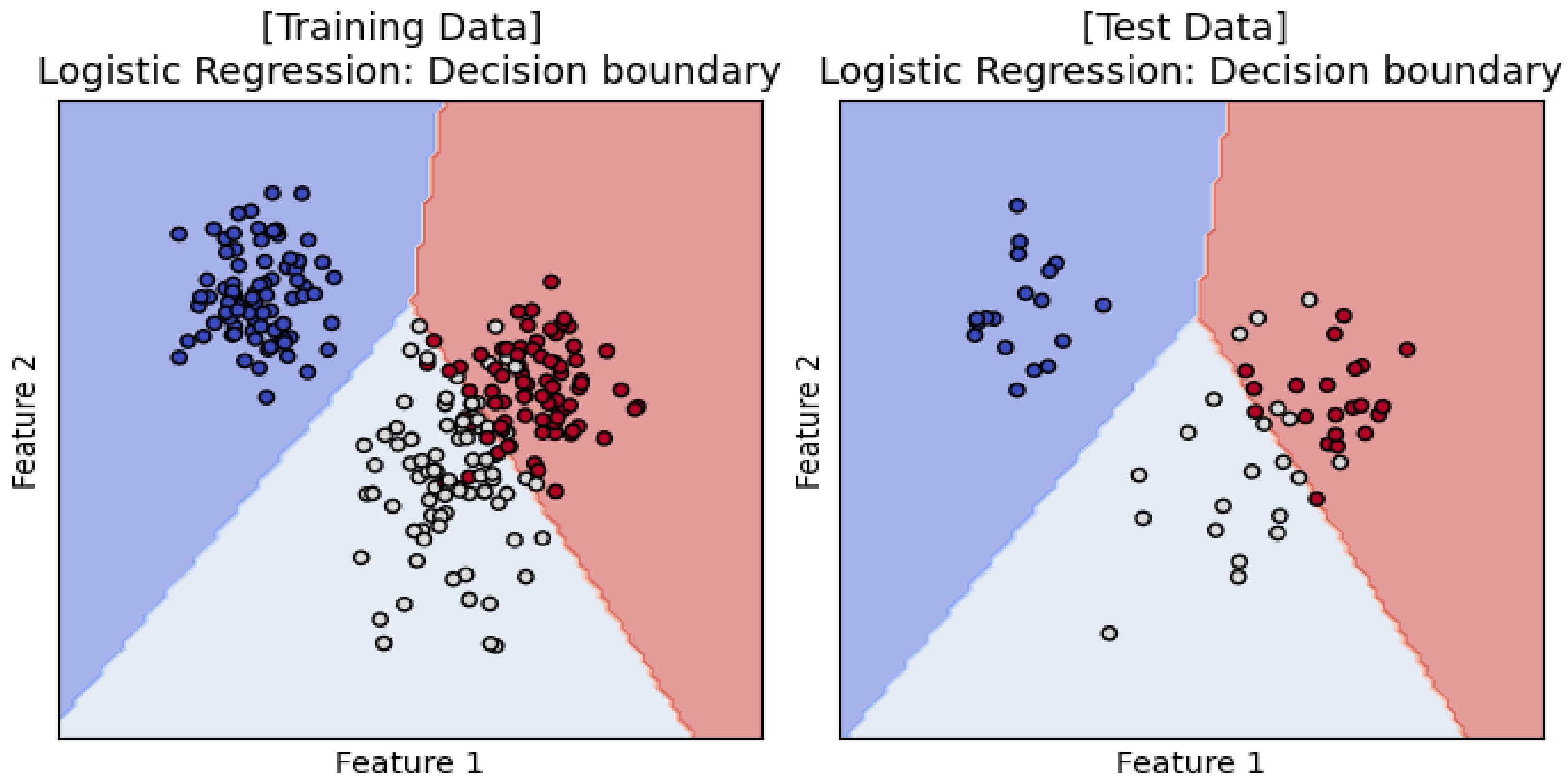
Logistic Regression



Though the small variations of decision boundaries are observed when using different solvers, the decision boundary of logistic regression exhibits linear behavior, showing a straight line.

Logistic Regression





Solvers ='lbfgs'

Decision boundary of logistic regression is consisting of <u>straight lines</u>.