# EN3150 Pattern Recognition
Classification
Part 01

M. T. U. Sampath K. Perera,
Department of Electronic and Telecommunication Engineering,
University of Moratuwa.
(sampathk@uom.lk).
Semester 5 – Batch 20.

Classification

➢ Linear regression assumes that dependent variable is quantitative*.

➢ Often, dependent variable is categorical variables (qualitative)

➢ Eye Color: Blue, Brown, Green.

➢ Weather Conditions: Sunny, Cloudy, Rainy, Snowy.

➢ Learning Environment: In-person, Remote, Hybrid.

➢ Numerical values can be converted to categorical by grouping numerical values into predefined ranges or bins.

➢ Temperature: Cold: <20°C, Mild: 20°C - 25°C, Warm: 26°C - 30°C, Hot: >30°C

➢ Exam marks: F < 35, C 35-50, B 50-75, A 75-100
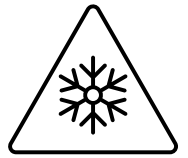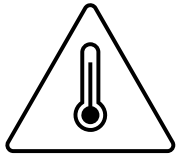
*data that can be measured and expressed numerically.

Classification

➢ What is classification?

➢ Process of categorizing data into distinct classes (categories) based on the characteristics or attributes of the data.

➢ In short "predicting qualitative responses"

classifying this observation
(data sample)
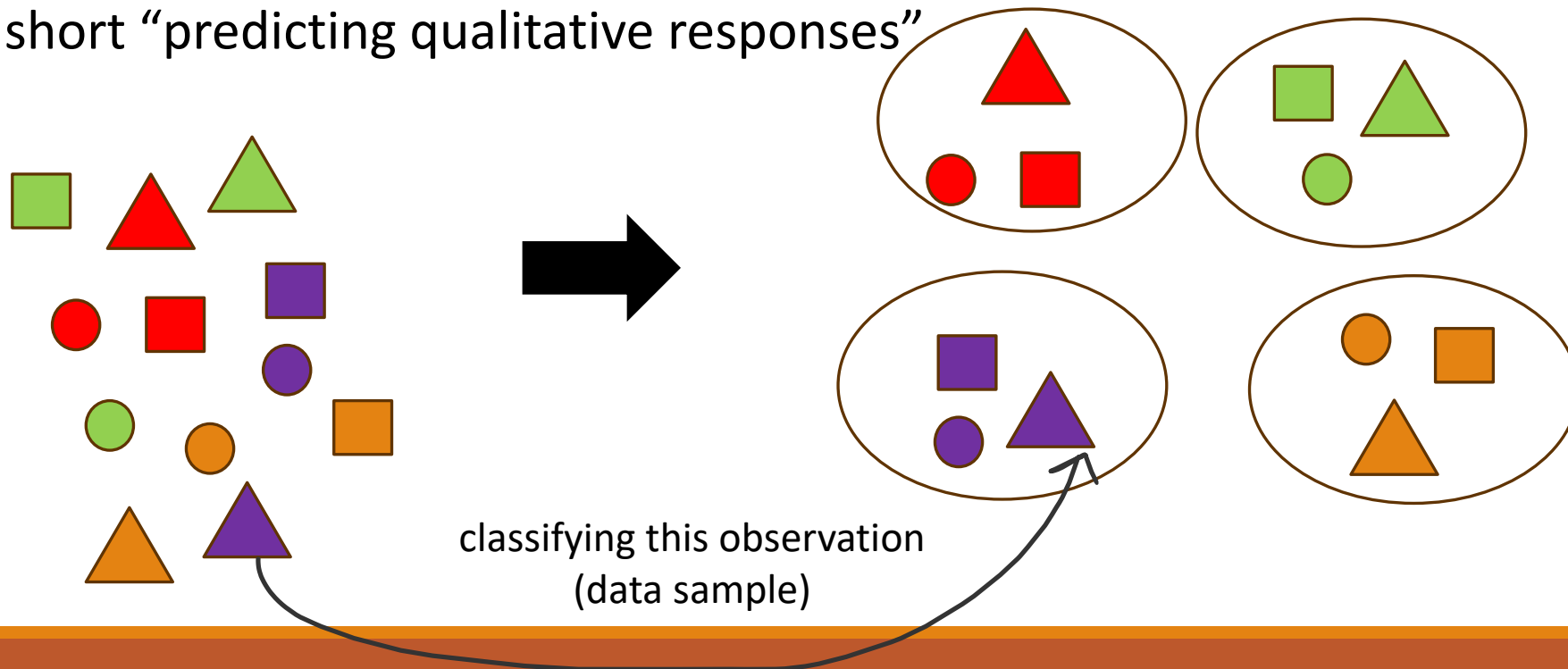
Classification or regression?
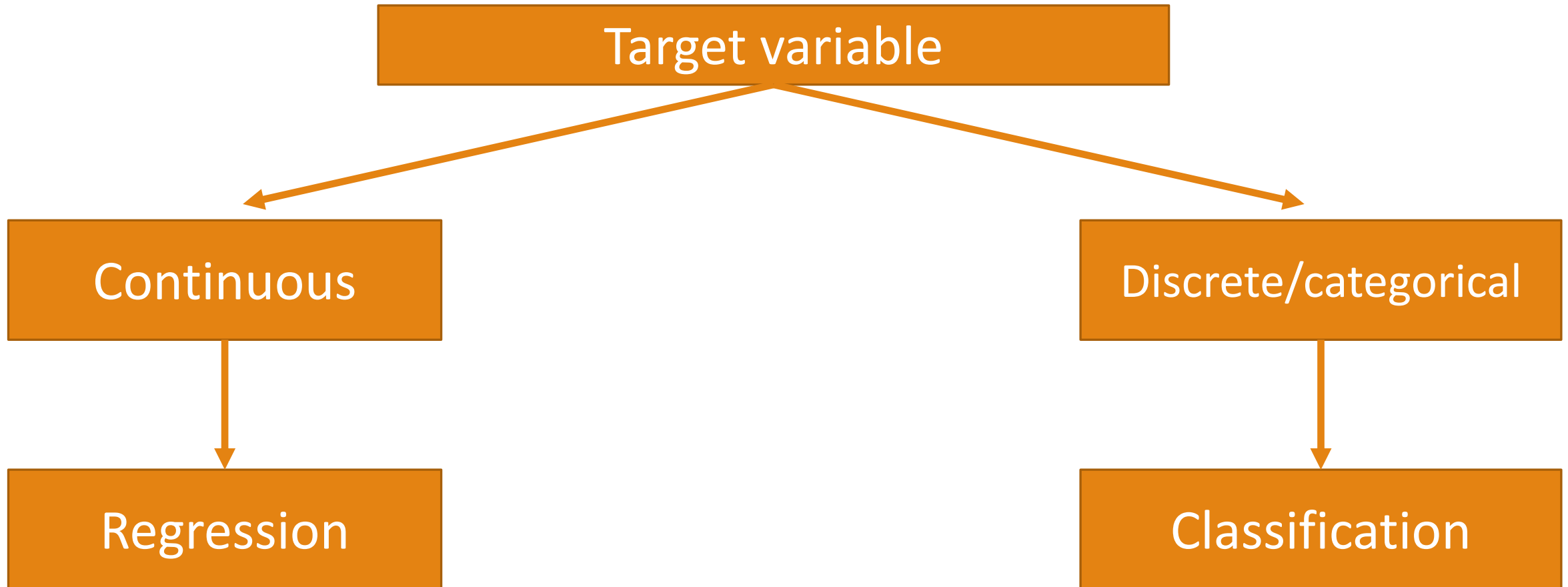
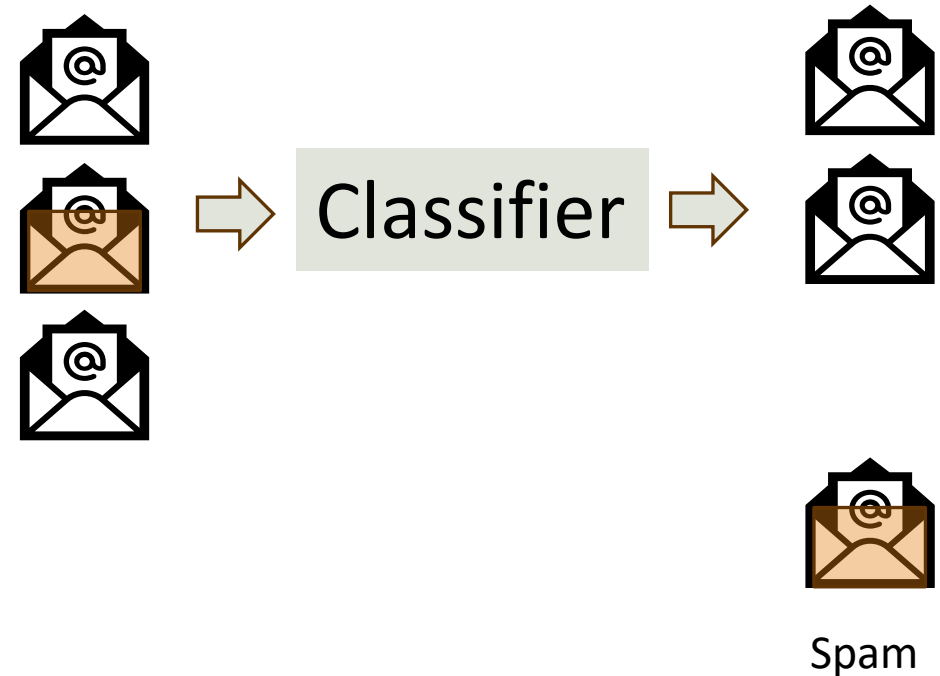# Classification

- Applications
  - Healthcare
  - Image classification
  - Document classification
  - Spam filtering
  - Facial recognition
  - Autonomous Vehicles

Classifier

Spam

Classification techniques or classifiers:

Logistic regression, linear discriminant analysis, K-nearest neighbors, trees, random forests, and boosting and naive bayes, neural networks, support vector machines.

Classification

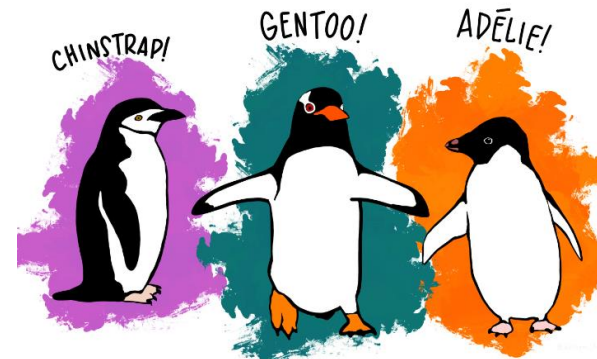➢Binary classification
  ➢Two distinct categories or classes
  ➢Algorithms: Logistic Regression, Support Vector Machines, Naive Bayes.

➢Multi-Class Classification
➢Three or more classes or categories
➢Algorithms*: Decision Trees, Random Forests, Neural Networks.
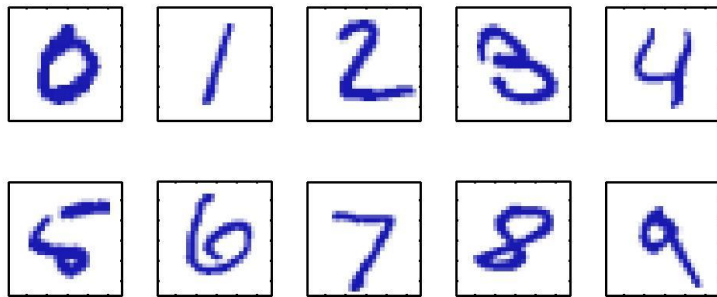
* Binary classifiers can be used with binary to multiclass conversion

Classification

➤ Binary to multiclass

  ➤ One-versus-one : Train many classifiers as there are pairs of labels

  ➤ One-versus-rest : Each class as an independent class and consider the rest combined as only one class

One-versus-one
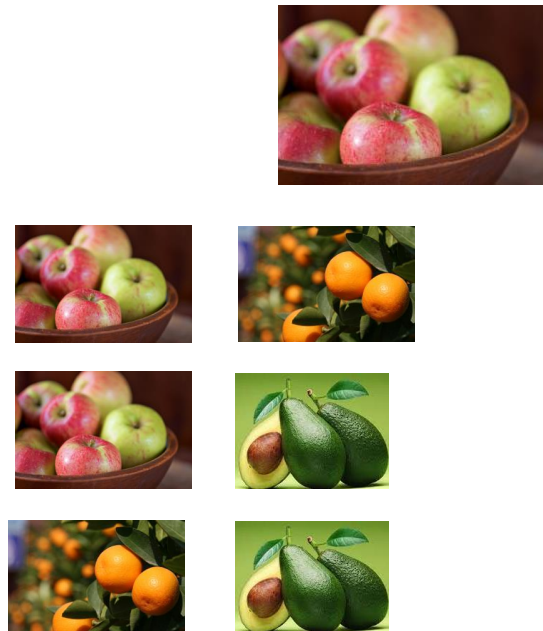
One-versus-rest (OVR)

N classes ➔ N x (N-1)/2 Binary Classifiers

N classes ➔ N Binary Classifiers

Classification and probability

➤ Probability provides a quantitative measure of the likelihood of a data point belonging to a particular class.
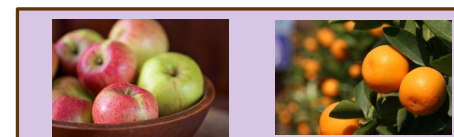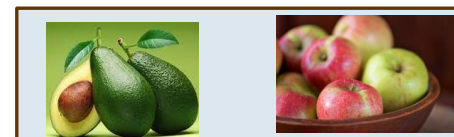
➤ "Classifiers first predict the probability of each of the categories of a qualitative variable, as the basis for making the classification"[1]

$p(C_1, |\mathbf{x}_1, \theta) = 0.1$ $C_1 -$ Spam class $p(C_k, |\mathbf{x}, \theta)$?

$p(C_1, |\mathbf{x}_2, \theta) = 0.05$

$p(C_1, |\mathbf{x}_3, \theta) = 0.01$

$p(C_1, |\mathbf{x}_4, \theta) = 0.7$

$\theta -$ Model parameters

[1] Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, 2013.

Linear models of classification

➤ Linear regression we assumed $p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|\mathbf{w}^T\mathbf{x}, \sigma^2)$

➤ Suppose that there are two classes (binary classification)

➤ What need to be change for $y \in \{0,1\}$? $p(y|\mathbf{x}, \mathbf{w}) = ?$ Bernoulli Distribution

➤ $y(\mathbf{x}) = w_0 + w_1 x_1 + \cdots, + w_D x_D = \mathbf{w}^T\mathbf{x}$

➤ Nonlinear function to convert $y(\mathbf{x})$*

$$p(y|\mathbf{x}, \mathbf{w}) = \text{Ber}(y|\text{f}(\mathbf{w}^T\mathbf{x}))$$
$$p(y|\mathbf{x}, \mathbf{w}) = \text{Ber}(y|\text{sigm}(\mathbf{w}^T\mathbf{x}))$$

$$\text{sigm}(\eta) \triangleq \frac{1}{1 + \exp(-\eta)} = \frac{e^\eta}{e^\eta + 1}$$

Sigmoid function

This is known as **Logistic regression**[Δ]
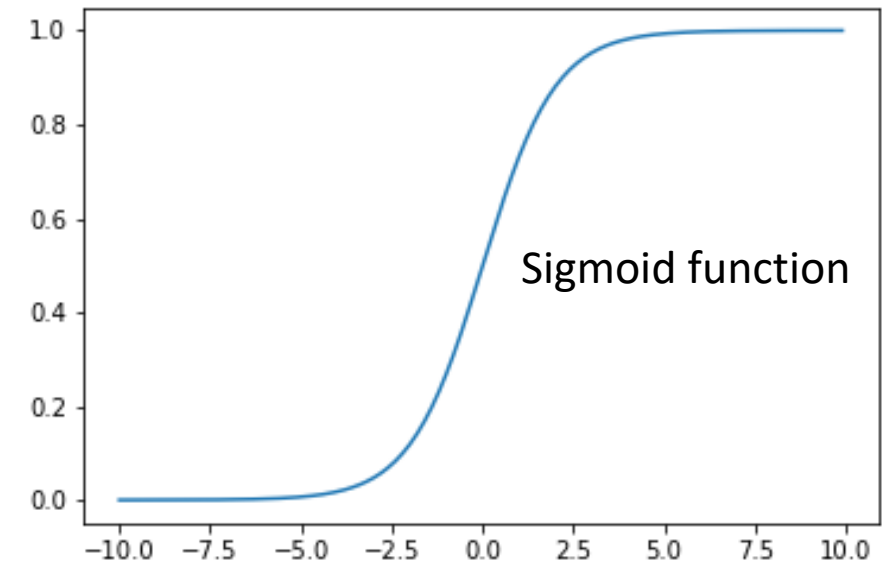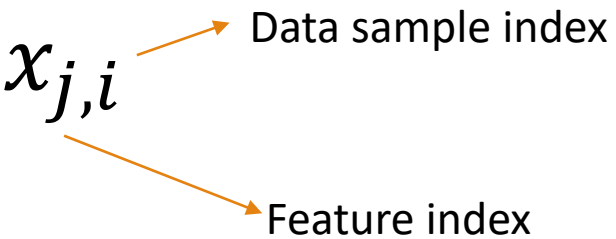
Linear models of classification

➤ For binary classification, we consider * $y_i \in \{0,1\} \; \forall i$

➤ Model output for $\mathrm{i-th}$ sample $y(\mathbf{x}_i) = w_0 + w_1\, x_{1,i} + \cdots, + w_D\, x_{D,i} = \mathbf{w}^T \mathbf{x}_i$

➤ Data set with N data samples

y	features
y_1	$x_{1,1}, x_{2,1}, ..., x_{D,1}$
y_2	$x_{1,2}, x_{2,2}, ..., x_{D,2}$
. . .	. . .
y_i	$x_{1,i}, x_{2,i}, ..., x_{D,i}$
y_N	$x_{1,N}, x_{2,N}, ..., x_{D,N}$

$$x_{j,i}$$

Data sample index

Feature index

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ . \\ . \\ . \\ w_D \end{bmatrix} \quad \mathbf{x}_i = \begin{bmatrix} 1 \\ x_{1,i} \\ . \\ . \\ . \\ x_{D,i} \end{bmatrix}$$

For single feature case, we drop the feature index to have a simple notation.
E.g., x_i - i-th sample

*Also, it is possible to use $y_i \in \{1, -1\} \forall i$

Linear models of classification

➢Categorical to numerical values*

   ➢One hot encoder

sklearn.preprocessing.OneH
otEncoder — scikit-learn
1.3.0 documentation

Data sample	Class label	One hot encoding		
1	Adelie	1	0	0
2	Gentoo	0	1	0
3	Chinstrap	0	0	1
4	Adelie	1	0	0

   ➢Integer encoding

Other encoding method see
6.3. Preprocessing data —
scikit-learn 1.3.0
documentation

Data sample	Class label	Integer encoding
1	Adelie	0
2	Gentoo	1
3	Chinstrap	2
4	Adelie	0

* Most algorithms it is required to convert class labels to numerical values

Linear models of classification

Sample Index (i)	Species (class) (y_i)	Island ($x_{1,i}$)	bill_length_mm ($x_{2,i}$)	bill_depth_mm ($x_{3,i}$)	flipper_length_mm ($x_{4,i}$)	body_mass_g ($x_{5,i}$)	sex ($x_{6,i}$)
1	Adelie	Torgersen	39.1	18.7	181.0	3750.00	Male
2	Chinstrap	Dream	46.5	17.9	192.0	3500.00	Female
3	Gentoo	Biscoe	46.1	13.2	211.0	4500.0	Female

➢Integer encoding

| Sample Index (i)	Species (class) (y_i)	Island ($x_{1,i}$)	bill_length_mm ($x_{2,i}$)	bill_depth_mm ($x_{3,i}$)	flipper_length_mm ($x_{4,i}$)	body_mass_g ($x_{5,i}$)	sex ($x_{6,i}$)
1	0	Torgersen	39.1	18.7	181.0	3750.00	Male
2	1	Dream	46.5	17.9	192.0	3500.00	Female
3	2	Biscoe	46.1	13.2	211.0	4500.0	Female

Linear models of classification

Sample Index (i)	Species (class) (y_i)	Island ($x_{1,i}$)	bill_length_mm ($x_{2,i}$)	bill_depth_mm ($x_{3,i}$)	flipper_length_mm ($x_{4,i}$)	body_mass_g ($x_{5,i}$)	sex ($x_{6,i}$)
1	Adelie	Torgersen	39.1	18.7	181.0	3750.00	Male
2	Chinstrap	Dream	46.5	17.9	192.0	3500.00	Female
3	Gentoo	Biscoe	46.1	13.2	211.0	4500.0	Female

➢One hot encoder

Sample Index (i)	Species (class) ($\mathbf{y}_i$)	Island ($x_{1,i}$)	bill_length_mm ($x_{2,i}$)	bill_depth_mm ($x_{3,i}$)	flipper_length_mm ($x_{4,i}$)	body_mass_g ($x_{5,i}$)	sex ($x_{6,i}$)
1	1 0 0	Torgersen	39.1	18.7	181.0	3750.00	Male
2	0 1 0	Dream	46.5	17.9	192.0	3500.00	Female
3	0 0 1	Biscoe	46.1	13.2	211.0	4500.0	Female

Linear models of classification

➢ For 1-D scenario (one feature x)
  ➢ $p(y_i = 1|x_i, \mathbf{w}) = \mathrm{sigm}(w_0 + w_1 x_i).$
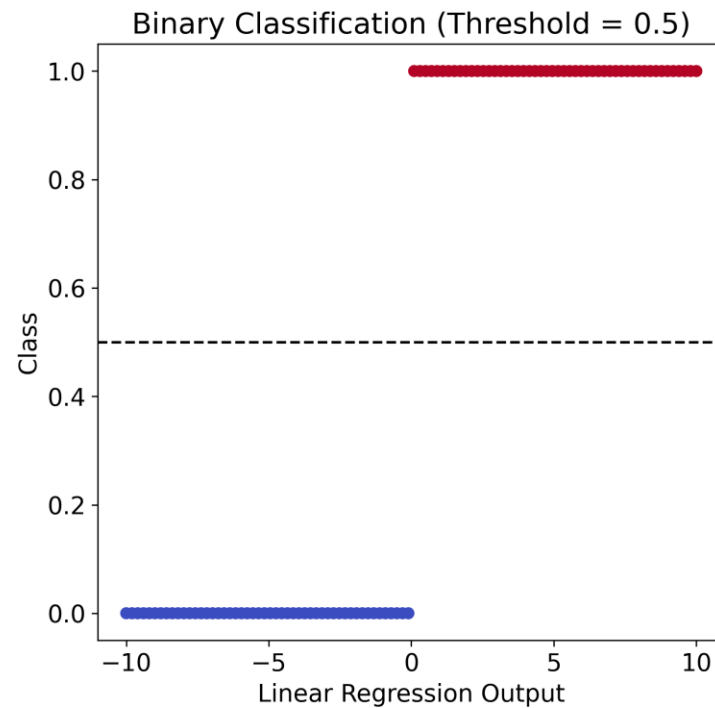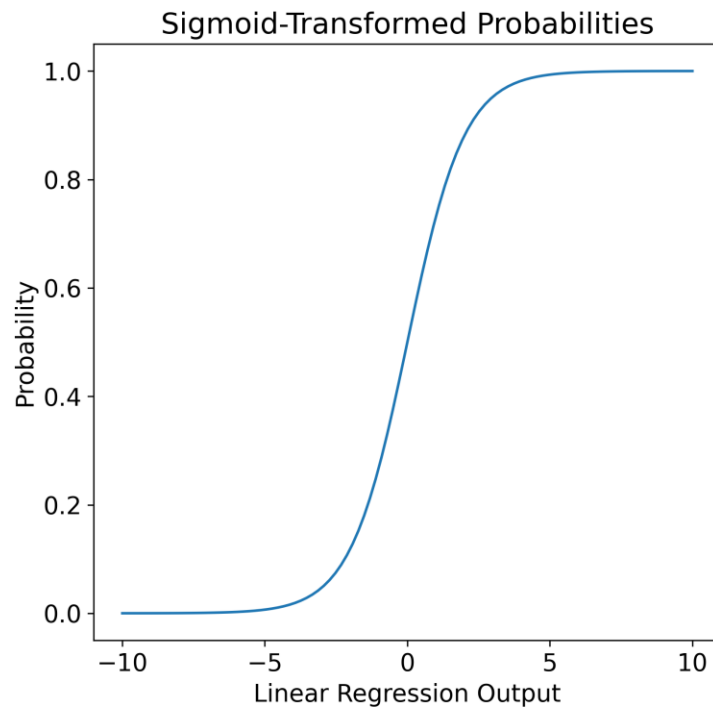
$\hat{y}(x) = 1 \iff p(y = 1|\mathbf{x}, \mathbf{w}) > 0.5$

$\hat{y}(x) = 0 \iff p(y = 1|\mathbf{x}, \mathbf{w}) < 0.5$

Linear models of classification

➢Sigmoid function, logistic, or logit function

$$\text{sigm}(\eta) \triangleq \frac{1}{1 + \exp(-\eta)} = \frac{e^\eta}{e^\eta + 1}$$

$$p(y_i = 1 | x_i, \mathbf{w}) = \text{sigm}(w_0 + w_1 x_i),$$ For convenient use the notation $p(y_i = 1 | x_i, \mathbf{w}) = p(x_i)$

$$p(x_i) = \frac{e^{w_0 + w_1 x_i}}{e^{w_0 + w_1 x_i} + 1}$$

Number of successes per failure= $\dfrac{p(x_i)}{1 - p(x_i)} = e^{w_0 + w_1 x_i}$ **Called odds**

➢What is the range of the odd?

The odds span from 0 to infinity, with an odds value of 0 indicating impossibility (p = 0), and an odds value of infinity indicating certainty (p = 1).

Linear models of classification

$\text{Log}_\text{e} \left(\frac{p(x_i)}{1 - p(x_i)} \right)$?

$\text{Log}_\text{e} \left(\frac{p(x_i)}{1 - p(x_i)} \right) = \text{Log}_\text{e} \left(e^{w_0 + w_1 x_i} \right) = w_0 + w_1 x_i$ **Called log-odds or logit**

➤ Linear predictor is obtained by taking a weighted sum of the predictor variables i.e., $w_0 + w_1 x_i$

➤ Can take any value on the real number line ➜ not suitable for representing probabilities.

➤ Logit transformation maps the linear predictor to a range between negative infinity and positive infinity. Next applying **the logistic function (sigmoid function)** to squash the values into the (0, 1) interval, which corresponds to valid probabilities.

$$Logit\big(p(x_i)\big) = w_0 + w_1 x_i$$

$$p(x_i) = \frac{e^{w_0 + w_1 x_i}}{e^{w_0 + w_1 x_i} + 1} = \frac{e^{Logit\big(p(x_i)\big)}}{e^{Logit\big(p(x_i)\big)} + 1}$$

produce probability estimates for binary outcomes based on linear combinations of predictor variables

Linear models of classification

➤How to learn the parameters $w_0, \ w_1, \dots, w_D$?

Due to the nonlinearity introduced by the logistic sigmoid function, logistic regression no longer has a closed-form solution.

$$\text{Log}_e \left(\frac{p(x_i)}{1-p(x_i)} \right) = w_0 + w_1 x_i \qquad p(x_i) = \frac{e^{w_0 + w_1 x_i}}{e^{w_0 + w_1 x_i} + 1} \qquad p(y|\mathbf{x}, \mathbf{w}) = \text{Ber}(y|\text{sigm}(\mathbf{w}^T \mathbf{x}))$$

➤Objective: To estimate $w_0, \ w_1$ to closely align the predicted probabilities $p(x_i)$ with the observed true class of individuals.

➤This can be translated into a likelihood function, which mathematically captures the probability of observing the data outcomes based on estimated parameters.

Linear models of classification

➢ Likelihood function

$$L(w_0, w_1) = \prod_{i, y_i = 1} p(x_i) \prod_{j, y_j = 0} (1 - p(x_j))$$

$$p(y_i = 1 | x_i, \mathbf{w}) = \text{sigm}(w_0 + w_1 x_i)$$
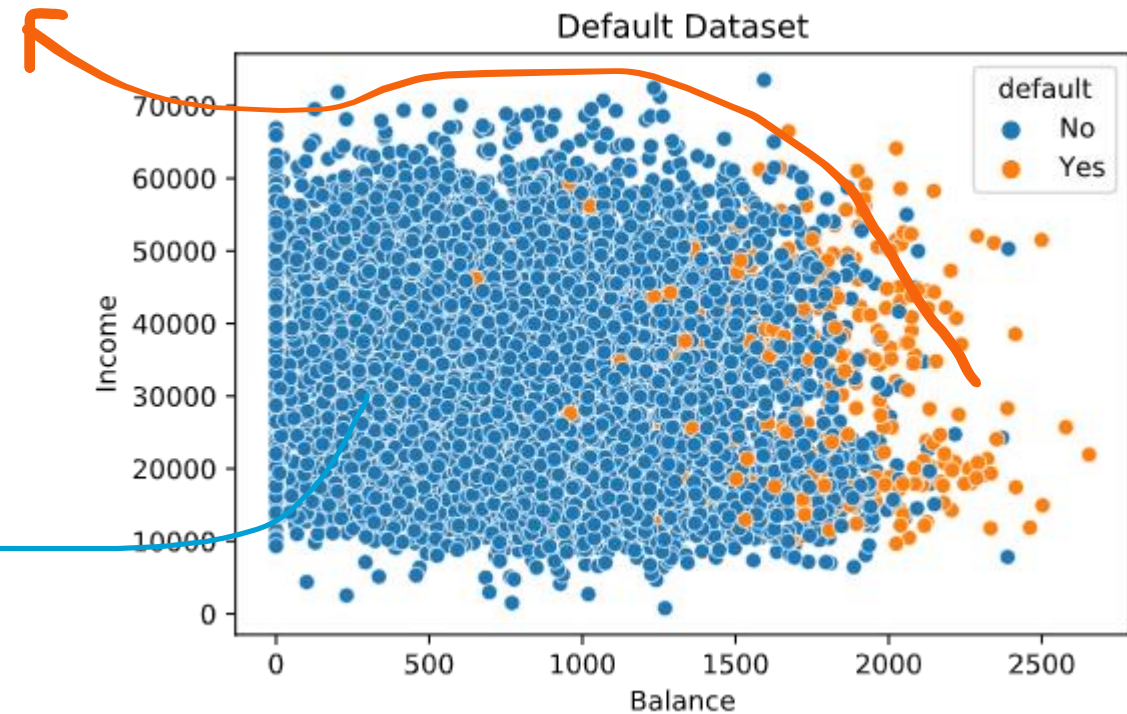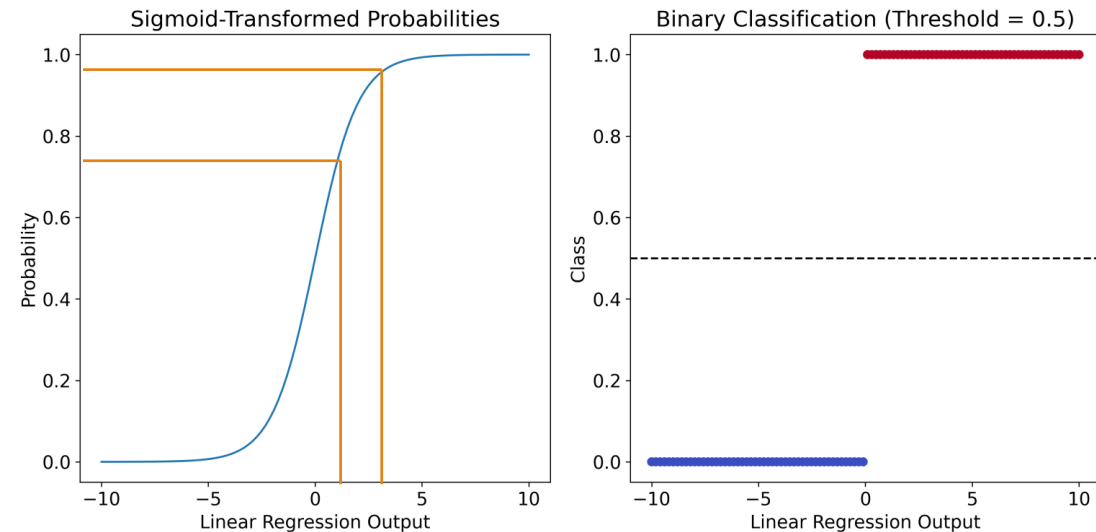
Linear models of classification

➤ Likelihood function
$$\text{L}(\mathbf{w}) = \prod_{i,y_i=1} p(x_i) \prod_{j,y_j=0} (1 - p(x_j))$$

$$p(y|\mathbf{x}, \mathbf{w}) = \text{Ber}n(y|\text{sigm}(\mathbf{w}^T\mathbf{x}))$$

$$
\begin{aligned}
\text{Bern}(x|\mu) &= \mu^x(1-\mu)^{1-x} \\
\mathbb{E}[x] &= \mu \\
\text{var}[x] &= \mu(1-\mu)
\end{aligned}
$$

➤ Negative log likelihood NLL($\mathbf{w}$)= - log L($\mathbf{w}$)

$$\text{NLL}(\mathbf{w}) = -\frac{1}{N}\log\prod_{n=1}^{N}\text{Bern}(y_i|\mu_i)$$

$$\text{NLL}(\mathbf{w}) = -\sum_{i=1}^{N}\log\left[\mu_i^{\mathbb{I}(y_i=1)} \times (1-\mu_i)^{\mathbb{I}(y_i=0)}\right]$$

$$\mu_i = \text{sigm}(\mathbf{w}^T\mathbf{x}_i),$$
this is probability of $\mathbf{x}_i$ for class 1

$$\text{NLL}(\mathbf{w}) = -\sum_{i=1}^{N}[y_i\log\mu_i + (1-y_i)\log(1-\mu_i)]$$

cross-entropy error function

$$\text{Binary Cross-Entropy} = -[y_i\log p + (1-y_i)\log(1-p)]$$

$\mathbb{I}(x)$- binary **indicator function.** Returns 1 iff (if and only if) the condition *x* is true, and returns 0 otherwise.

Linear models of classification

➤ For $y \in \{1, -1\}$

$$\text{NLL}(\mathbf{w}) = -\frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i = 1)\log[\text{sigm}(\mathbf{w}^T\mathbf{x}_i)] + \mathbb{I}(y_i = -1)\log[\text{sigm}(-\mathbf{w}^T\mathbf{x}_i)]$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \log[\text{sigm}(\widetilde{y}_i \mathbf{w}^T\mathbf{x}_i)]$$

$$= -\frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i \mathbf{w}^T\mathbf{x}_i))$$

$$\text{sigm}(-\mathbf{w}^T\mathbf{x}_i) = 1 - \text{sigm}(\mathbf{w}^T\mathbf{x}_i)$$

$\mathbb{I}(x)$- binary **indicator function.** Returns 1 iff (if and only if) the condition *x* is true and returns 0 otherwise.

Linear models of classification

➢How to estimate $\mathbf{w}$?

➢$\nabla_{\mathbf{w}}$ NLL ($\mathbf{w}$)=g($\mathbf{w}$)=0 optimization algorithm (gradient based) can be used to solve this.

$$\text{NLL}(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N} [y_i \log \text{sigm}(\mathbf{w}^T\mathbf{x}_i) + (1-y_i)\log(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i))]$$

Task: What is g($\mathbf{w}$) ?

$$\text{sigm}(a) \triangleq \frac{1}{1+\exp(-a)} = \frac{e^a}{e^a+1} \qquad \frac{\partial \, \text{sigm}(a)}{\partial a} = \text{sigm}(a)\,(1-\text{sigm}(a))$$

$$\frac{\partial \mathbf{w}^T\mathbf{x}_i}{\partial \mathbf{w}} = \mathbf{x}_i \qquad \frac{\partial \, \text{sigm}(\mathbf{w}^T\mathbf{x}_i)}{\partial \mathbf{w}} = \frac{\partial \, sigm(a)}{\partial \mathbf{w}} = \left(\frac{\partial a}{\partial \mathbf{w}}\right)\left(\frac{\partial \, sigm(a)}{\partial a}\right) = (\mathbf{x_i})(\text{sigm}(a)\,(1-\text{sigm}(a)))$$

$$= (\mathbf{x_i})(\text{sigm}(\mathbf{w}^T\mathbf{x}_i)\,(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i)))$$

Linear models of classification

$$\text{NLL}(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N} \quad [y_i \log \text{sigm}(\mathbf{w}^T\mathbf{x}_i) + (1-y_i)\log(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i))]$$

$$\frac{\partial(\log \text{sigm}(\mathbf{w}^T\mathbf{x}_i))}{\partial \mathbf{w}} = \frac{1}{\text{sigm}(\mathbf{w}^T\mathbf{x}_i)}\frac{\partial(\quad \text{sigm}(\mathbf{w}^T\mathbf{x}_i))}{\partial \mathbf{w}} = \frac{1}{\text{sigm}(\mathbf{w}^T\mathbf{x}_i)}(\mathbf{x}_i)(\text{sigm}(\mathbf{w}^T\mathbf{x}_i)\,(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i))$$

$$= (\mathbf{x}_i)(1\text{- }\text{sigm}(\mathbf{w}^T\mathbf{x}_i))$$

$$\frac{\partial(\log(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i))}{\partial \mathbf{w}} = \frac{1}{(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i)}\frac{\partial((1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i))}{\partial \mathbf{w}}$$

$$= \frac{1}{(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i)}(-\mathbf{x}_i)\,(\text{sigm}(\mathbf{w}^T\mathbf{x}_i)\,(1-\text{sigm}(\mathbf{w}^T\mathbf{x}_i)) = (-\mathbf{x}_i)\,(\text{sigm}(\mathbf{w}^T\mathbf{x}_i)$$

$$\nabla_{\mathbf{w}}\text{NLL}(\mathbf{w}) = g(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N} y_i\,(\mathbf{x}_i)(1\text{- }\text{sigm}(\mathbf{w}^T\mathbf{x}_i)) - (1-y_i)(\mathbf{x}_i)\,(\text{sigm}(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N}(\text{sigm}(\mathbf{w}^T\mathbf{x}_i)-y_i)\mathbf{x}_i$$

Linear models of classification

➢ Stochastic gradient descent

➢ Update of weight (t is the sample index)

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \alpha \, \nabla_{\mathbf{w}} \, \text{NLL} \, (\mathbf{w}_t)$$

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \alpha \, \boxed{(\text{sigm}(\mathbf{w}^T \mathbf{x}_t) - y_t)} \, \mathbf{x}_i$$

➢ Batch gradient descent

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \frac{1}{N} \sum_{i=1}^{N} \boxed{(\text{sigm}(\mathbf{w}^T \mathbf{x}_i) - y_i)} \, \mathbf{x}_i$$

Linear models of classification

➤ How we guaranteed that we get global minimum?
  ➤ Gradient-based optimizers find points where the gradient is zero
  ➤ Such points can be global or local optima.
  ➤ To confirm a global optimum, the objective function's convexity is essential.
  ➤ Positive semi-definiteness of the Hessian matrix ensures that the function is convex.
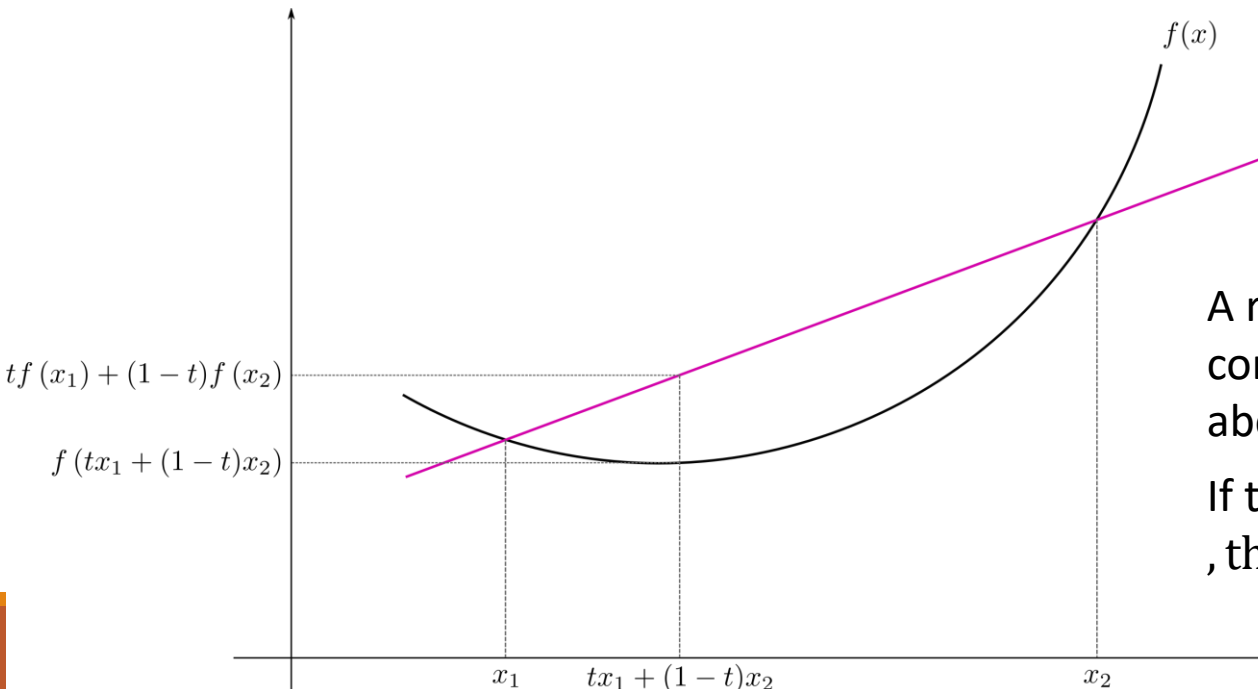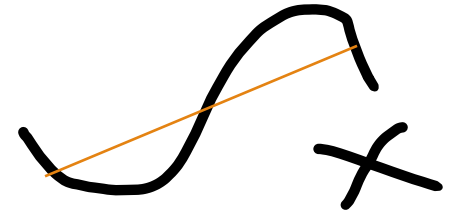
$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$$

A real valued function is said convex if the line segment connecting any two different points on its graph is positioned above the graph's curve within those two points.

If the inequality holds strictly, i.e., $<$ than $\leq$, then f() is stircly convex function.

Linear models of classification

➤ How we guaranteed that we get global minimum?
  ➤ Positive semi-definiteness of the Hessian matrix ensures that the function is convex.
  ➤ Hessian is given by

$$\nabla_{\mathbf{w}} \text{ NLL }(\mathbf{w})=g(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} (\text{sigm}(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{x}_i = \frac{1}{N} \mathbf{1}^T{}_N (\text{diag}(\text{sigm}(\mathbf{w}^T\mathbf{x}_i) - y_i)X)^T$$

$$\nabla_{\mathbf{w}} \nabla^{\mathbf{T}}_{\mathbf{w}} \text{ NLL }(\mathbf{w})= \frac{1}{N}\sum_{i=1}^{N} \text{sigm}(\mathbf{w}^T\mathbf{x}_i)(1- \text{sigm}(\mathbf{w}^T\mathbf{x_i}))(\mathbf{x}_i)(\mathbf{x}^T{}_i) = \frac{1}{N} X^TSX$$

$$S = \text{diag}(\text{sigm}(\mathbf{w}^T\mathbf{x}_1)(1- \text{sigm}(\mathbf{w}^T\mathbf{x}_1), \text{sigm}(\mathbf{w}^T\mathbf{x}_2)(1- \text{sigm}(\mathbf{w}^T\mathbf{x}_2), \dots, \text{sigm}(\mathbf{w}^T\mathbf{x}_N)(1- \text{sigm}(\mathbf{w}^T\mathbf{x}_N))$$

$$v^TX^TSXv= v^TX^T S^{\frac{1}{2}} S^{\frac{1}{2}} Xv = \left\| v^TX^T S^{\frac{1}{2}} \right\|^2 > 0* \text{ Here, } v \text{ is any non-zero vector}$$

Linear models of classification

➢ Gradient descent : challenges

➢ Gradient descent is a first order optimization method, can be "slow" convergence.

"This can be slow, especially when some directions of space point steeply downhill, whereas other have a shallower gradient[1]"

➢ Second-Order Methods: Use second-order information (Hessian matrix) in addition to gradients. Further, these methods consider curvature of the space. They can converge faster but are often computationally more expensive.

[1] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.

Linear models of classification

➤ Newton's method (Newton–Raphson method)

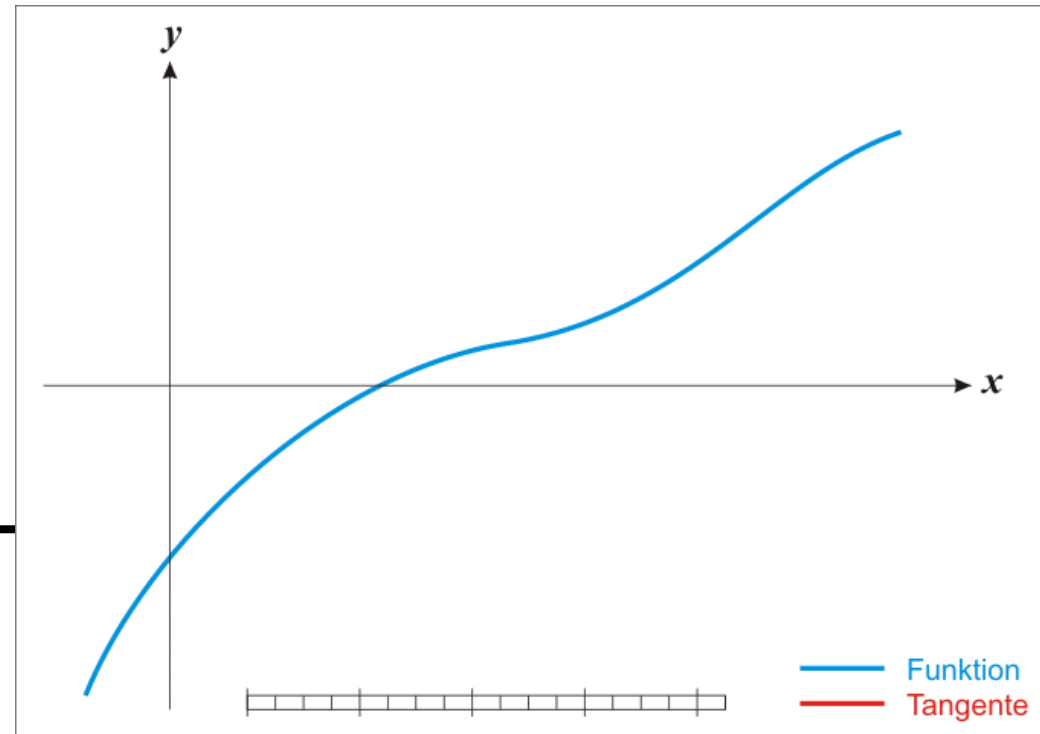➤ Objective to find x with $f(x)$=0

$f(x)$

$f(x_0)$
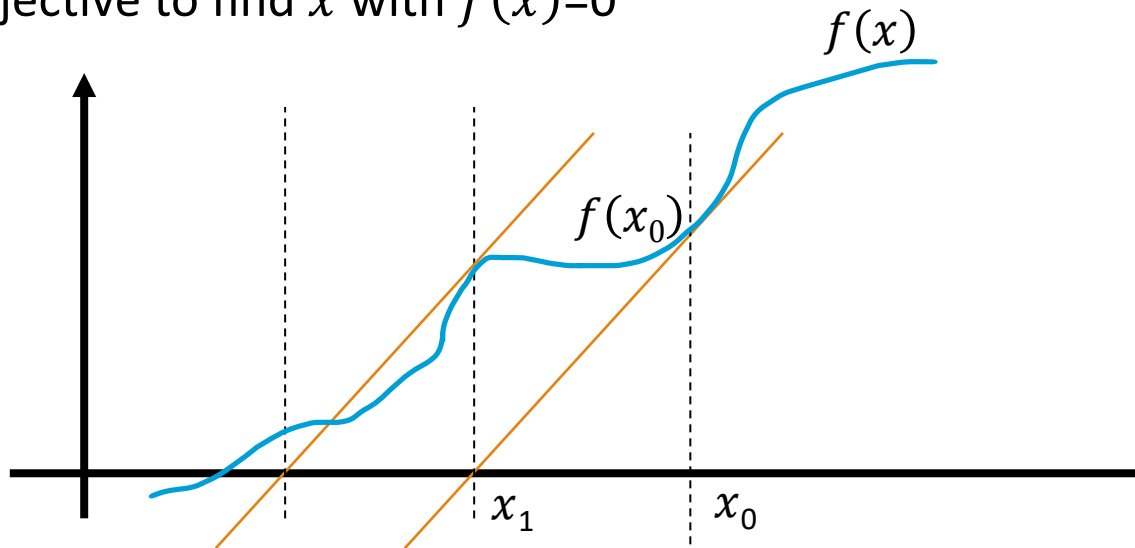
x_1 x_0

$f'(x_0)=f(x_0)/(x_0-x_1)$ ➔ $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)}$ let $g'(x)= f(x)$ $x_{t+1} = x_t - \frac{g'(x_t)}{g''(x_t)} = x_t - (g''(x_t))^{-1} g'(x_t)$

Animation from Newton's method - Wikipedia

Linear models of classification

➢ Newton's method

➢ $\boldsymbol{w}_{new} \leftarrow \boldsymbol{w_{old}} - \alpha \, \mathrm{H}^{-1} \nabla \mathrm{E}\,(\mathbf{w})$

➢ H - Hessian matrix of E $(\mathbf{w})$

➢ If Hessian is exact $\alpha=1$

A positive-definite Hessian matrix is required to ensure that the optimization update is well-defined and converges towards the minimum of the function

$$\nabla_{\mathbf{w}} \, \mathrm{NLL}\,(\mathbf{w}) = g(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\,(\mathrm{sigm}(\mathbf{w}^T\mathbf{x}_i) - y_i\,)\mathbf{x}_i$$

$$\nabla_{\mathbf{w}}\,\nabla^{\mathbf{T}}_{\mathbf{w}}\,\mathrm{NLL}\,(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\mathrm{sigm}(\mathbf{w}^T\mathbf{x}_i)\,(1-\mathrm{sigm}(\mathbf{w}^T\mathbf{x}_i))\,(\mathbf{x}_i)\,(\mathbf{x}^{\mathrm{T}}_i) = \frac{1}{N}\,\boldsymbol{X^T S X}$$

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \alpha\,\mathrm{H_t}^{-1}\nabla\mathrm{E}\,(\boldsymbol{w}_t)$$

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + (\boldsymbol{X^T S_t X})^{-1}(y_t - \mathrm{sigm}(\boldsymbol{w}_t{}^T\mathbf{x}_t))\mathbf{x}_t$$

$$\boldsymbol{w}_{t+1} \leftarrow [(\boldsymbol{X^T S_t X})^{-1}][(\boldsymbol{X^T S_t X})\boldsymbol{w}_t + (y_t - \mathrm{sigm}(\boldsymbol{w}_t{}^T\mathbf{x}_t))\mathbf{x}_t]$$

An "exact Hessian" refers to the Hessian matrix that is computed directly from the second-order partial derivatives of a function.

Linear models of classification

$\nabla_{\mathbf{w}}$ NLL $(\mathbf{w})$=$g(\mathbf{w})$ = $\frac{1}{N}\sum_{i=1}^{N}$ $(\text{sigm}(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{x}_i$= $\frac{1}{N}$ $\mathbf{1}^T{}_N$ $(\text{diag}(\text{sigm}(\mathbf{w}^T\mathbf{x}_i) - y_i)\mathbf{X})^T$

For one sample $\nabla_{\mathbf{w}}$ NLL $(\mathbf{w})$=$g(\mathbf{w})$ = $\mathbf{X}^T$ $(\text{sigm}(\mathbf{w}^T\mathbf{x}_i) - y_i)$ = $\mathbf{X}^T$ $(\mu_i - y_i)$, $where$ μ_i = $\text{sigm}(\mathbf{w}^T\mathbf{x}_i)$.

$\mathbf{w}_{t+1} \leftarrow [(\mathbf{X}^T\mathbf{S}\mathbf{X})^{-1}][(\mathbf{X}^T\mathbf{S}_t\mathbf{X})\mathbf{w}_t + \mathbf{X}^T(y_t - \mu_t)]$

$\mathbf{w}_{t+1} \leftarrow [(\mathbf{X}^T\mathbf{S}\mathbf{X})^{-1}]\mathbf{X}^T[(\mathbf{S}_t\mathbf{X})\mathbf{w}_t + (y_t - \mu_t)]$

$\mathbf{w}_{t+1} \leftarrow [(\mathbf{X}^T\mathbf{S}_t\mathbf{X})^{-1}]\mathbf{X}^T\mathbf{S}_t\mathbf{Z}_t$ Where, $\mathbf{Z}_t$ = $\mathbf{X}\mathbf{w}_t + (\mathbf{S}_t)^{-1} + (y_t - \mu_t)$

$\mathbf{S}$ = $\text{diag}(\text{sigm}(\mathbf{w}^T\mathbf{x}_1)(1- \text{sigm}(\mathbf{w}^T\mathbf{x}_1), \text{sigm}(\mathbf{w}^T\mathbf{x}_2)(1- \text{sigm}(\mathbf{w}^T\mathbf{x}_2), \ldots, \text{sigm}(\mathbf{w}^T\mathbf{x}_N)(1- \text{sigm}(\mathbf{w}^T\mathbf{x}_N))$

$\mathbf{S}_t$ = $\text{diag}(\text{sigm}(\mathbf{w}_t{}^T\mathbf{x}_1)(1- \text{sigm}(\mathbf{w}_t{}^T\mathbf{x}_1), \text{sigm}(\mathbf{w}_t{}^T\mathbf{x}_2)(1- \text{sigm}(\mathbf{w}_t{}^T\mathbf{x}_2), \ldots, \text{sigm}(\mathbf{w}_t{}^T\mathbf{x}_N)(1- \text{sigm}(\mathbf{w}_t{}^T\mathbf{x}_N))$

Linear models of classification

$$w_{t+1} \leftarrow \left[(X^T S_t X)^{-1} \right] X^T S_t Z_t$$

$$Z_t = X w_t + (S_t)^{-1} + \left(y_t - \mu_t \right)$$

$$z_{t,n} = \mathbf{w}_t^T \mathbf{x}_n + \frac{y_n - \mu_{t,n}}{\mu_{t,n} (1 - \mu_{t,n})}$$

➢ Linear regression or ordinary least squares: $\widehat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$

➢ Weighted least squares: $\widehat{\mathbf{w}}_{\text{wLS}} = (\mathbf{X}^T \mathbf{\Lambda} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{\Lambda} \mathbf{y}$.

➢ There is an associate a weight with each example.

➢ Loss function* $\sum_{i=1}^{N} a_i (y_i - \mathbf{w}^T \mathbf{x}_i)^2$

Rubin, D. B. (1983). Iteratively reweighted least squares. In *Encyclopedia of Statistical Sciences*, Volume 4, pp. 272–275. Wiley.

* Here, these weights (a_i) need to be design. One approach is to select them inverse of the variance of the observations. Data points with higher reliability or smaller measurement errors would be assigned larger weights in weighted least squares. This adjustment signifies that these particular data points have a more significant impact on shaping the regression model during the fitting process.

Linear models of classification

Iteratively reweighted least squares (IRLS)

Algorithm 1 Iteratively Reweighted Least Squares (IRLS)

1: Initialize weights $\mathbf{w} = \mathbf{0}$
2: **repeat**
3:     **for** $n = 1$ to N **do**
4:         $a_n = \mathbf{w}^{\mathbf{T}}\mathbf{x_n}$
5:         $\mu_n = \mathbf{sigm}(a_n)$
6:         $s_n = \mu_n(1 - \mu_n)$
7:         $z_n = a_n + \frac{y_n - \mu_n}{s_n}$
8:     **end for**
9:
10:     $S = \mathbf{diag}(s_1, s_2, \ldots, s_N)$
11:     Update weights: $\mathbf{w} = (\mathbf{X^T S X})^{-1}\mathbf{X^T S z}$
12: **until** converged

Rubin, D. B. (1983). Iteratively reweighted least squares. In *Encyclopedia of Statistical Sciences*, Volume 4, pp. 272–275.Wiley.
Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.

Linear models of classification

➢Logistic Regression $\text{NLL}(\mathbf{w}) = -\sum_{i=1}^{N} [y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)]$ for $y_i \in \{0, 1\}$

➢Logistic regression with regularization

$$\text{NLL}(\mathbf{w}) = C \sum_{i=1}^{N} -[y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)] + r(\mathbf{w})$$

➢L2 penalty $r(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_2^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w}$

➢L1 penalty $r(\mathbf{w}) = \|\mathbf{w}\|_1$

➢ElasticNet $r(\mathbf{w}) = \frac{1-\rho}{2} \mathbf{w}^T \mathbf{w} + \rho \|\mathbf{w}\|_1$

$$\mathbf{v} = v_1, v_2, \ldots, v_n$$

$$\|\mathbf{v}\|_1 = |v_1| + |v_2| + \cdots + |v_n|$$

$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$$

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

Greater values of C provide the model with increased flexibility, while smaller values of C impose stricter constraints on the model.

Linear models of classification

Logistic regression is widely favored for several reasons:

➢ its simplicity in fitting (straightforward to implement).

➢ Fast compared to Support Vector Machines (SVMs), Neural Networks (NNs), and Decision Trees.

➢ Easy to interpret.

➢ It can be extended to multi-class classification.

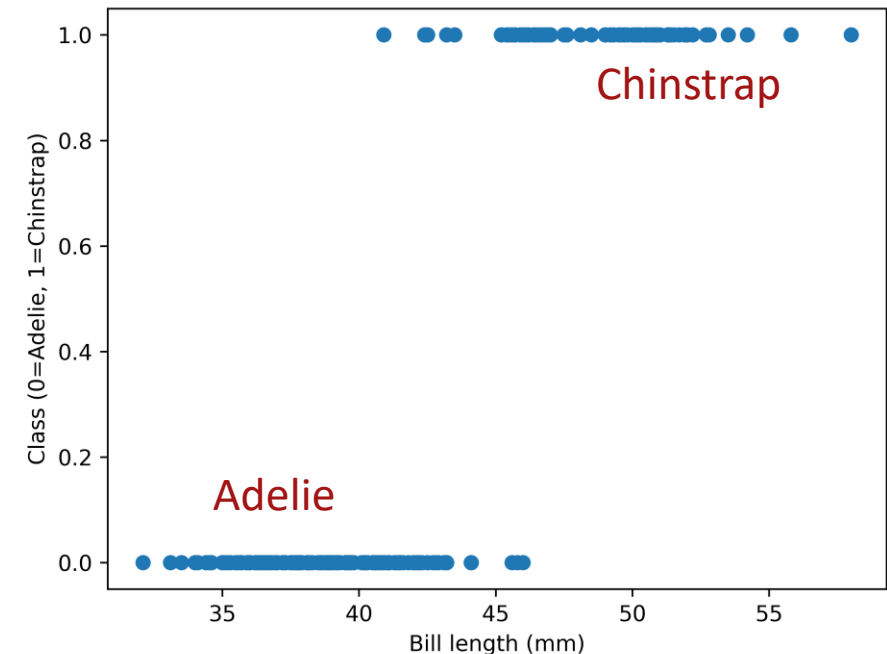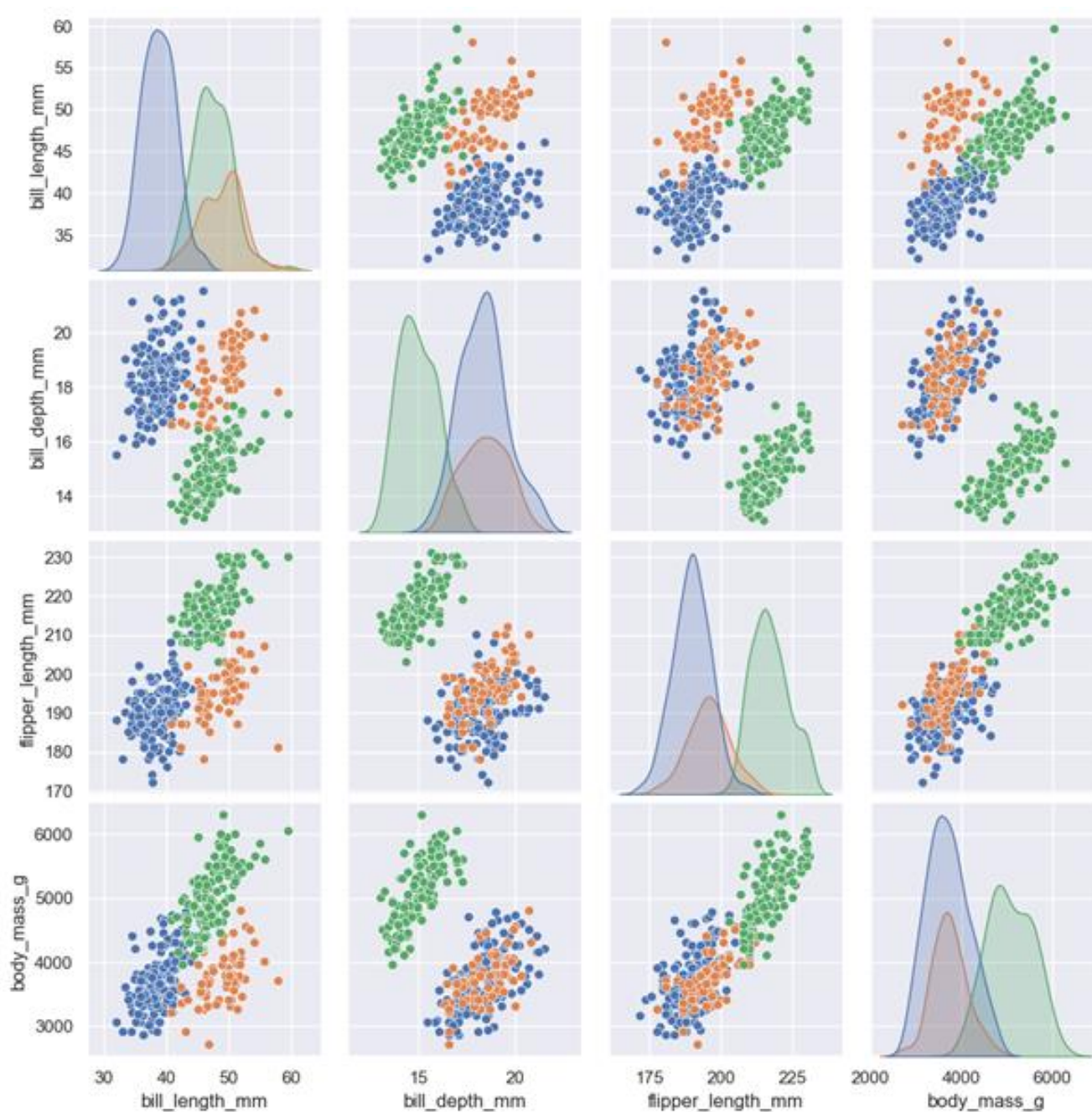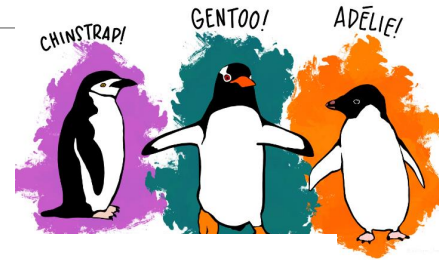➢ It can be extended to accommodate non-linear decision boundaries by employing kernel functions.

Linear models of classification

Feature='Bill length'
Classes:
0=Adelie, 1=Chinstrap

Chinstrap

Adelie

Linear models of classification

➢ $p(y_i = 1 | x_i, \mathbf{w}) = \text{sigm}(w_0 + w_1 x_i)$

➢ After learning w_0 and w_1 red curve can be plotted.

The point at which the sigmoid function $\text{sigm}(w_0 + w_1 x_i) = 0.5$ for $x^* \approx 44.10$ is satisfied can be identified as the decision boundary.

For the specific feature value x^* around 44.10, this boundary distinguishes between two classes.

On the left side, instances are classified as 0, while on the right side, instances are classified as 1.
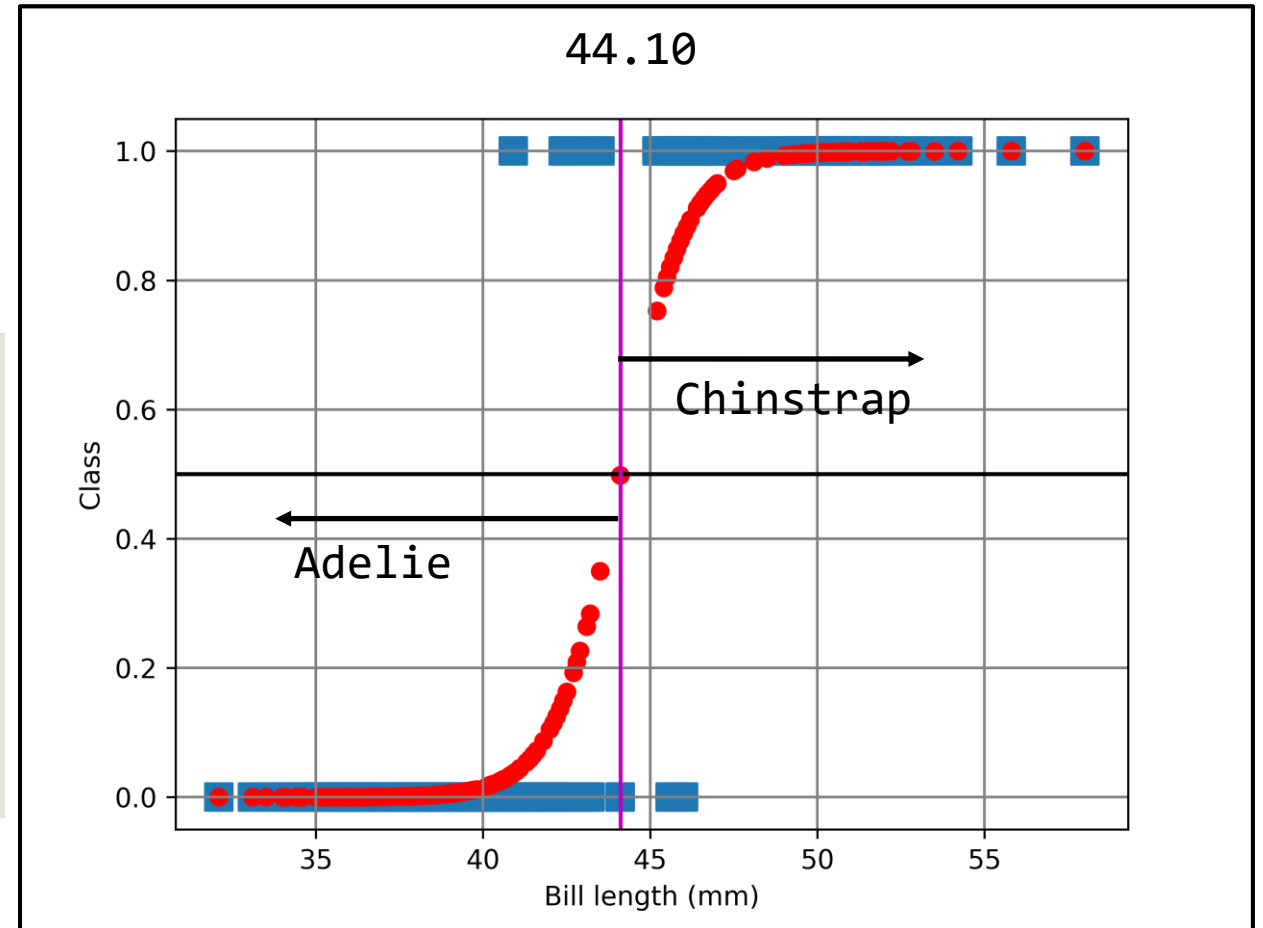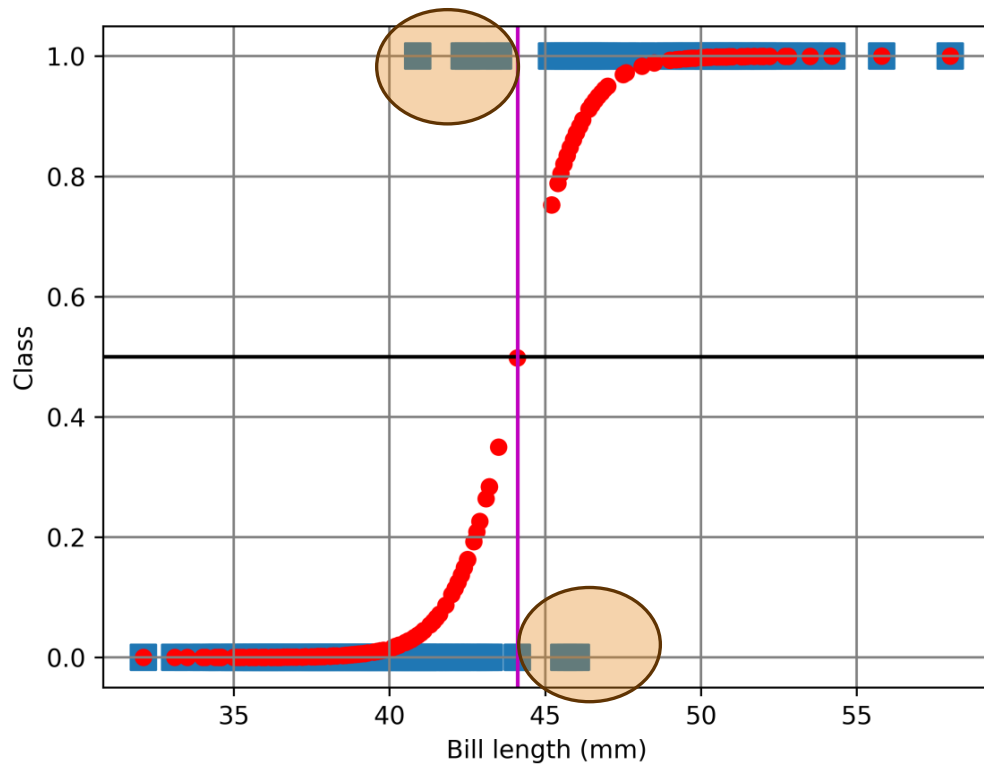
Linear models of classification

➤ This decision criterion yields an error rate that isn't zero even when applied to the training dataset.

➤ There isn't a straight line that can be drawn to distinctly separate the instances categorized as 0 from those categorized as 1.

➤ To address this, we can formulate models featuring non-linear decision boundaries by employing basis function expansion, similar to our approach with non-linear regression.
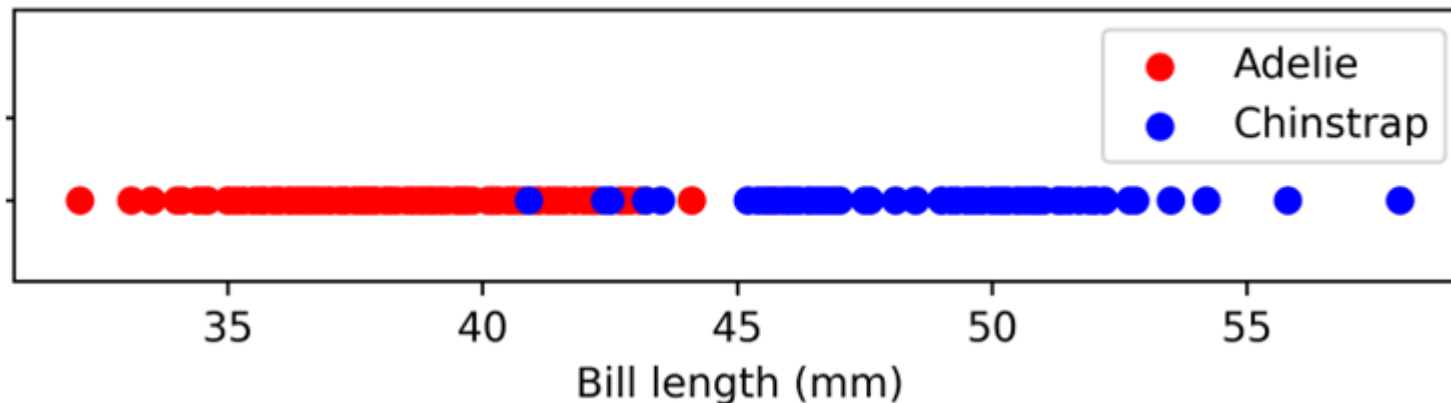
Linear models of classification

$$p(x_i) = \frac{e^{w_0 + w_1 x_i}}{e^{w_0 + w_1 x_i + 1}}$$

w_1 = 1.02, w_0 =-44.91

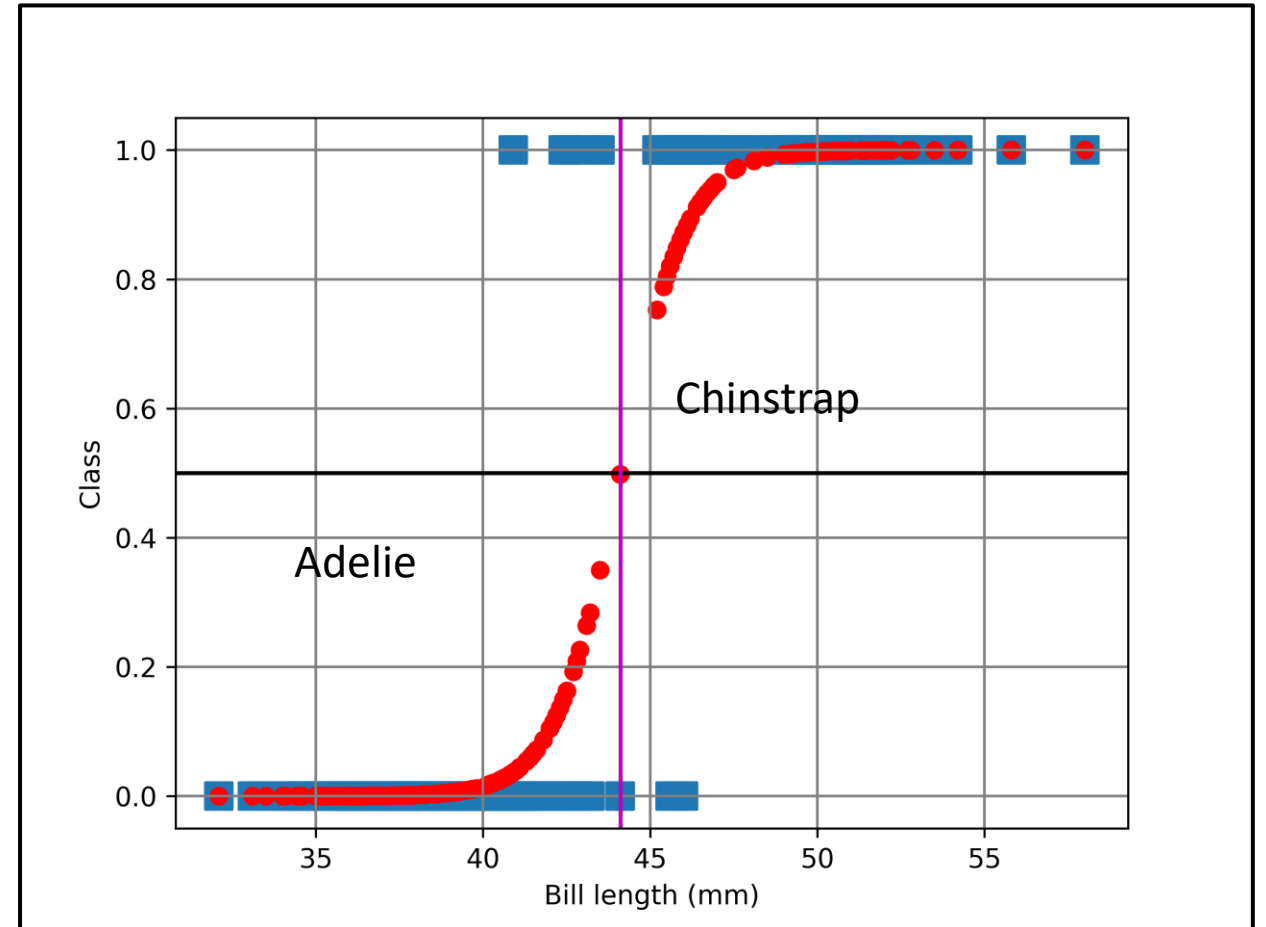Prediction
bill length (x_i) = 32 mm, 42 mm and 55 mm

$$p(x_i) = \frac{e^{-44.91 + 1.02 x_i}}{e^{e^{-44.91 + 1.02 x_i}} + 1}$$

$p(32) =$	4.69e-06	Adelie
$p(42) =$	0.11	Adelie
$p(55) =$	0.99	Chinstrap

Default Dataset

Linear models of classification
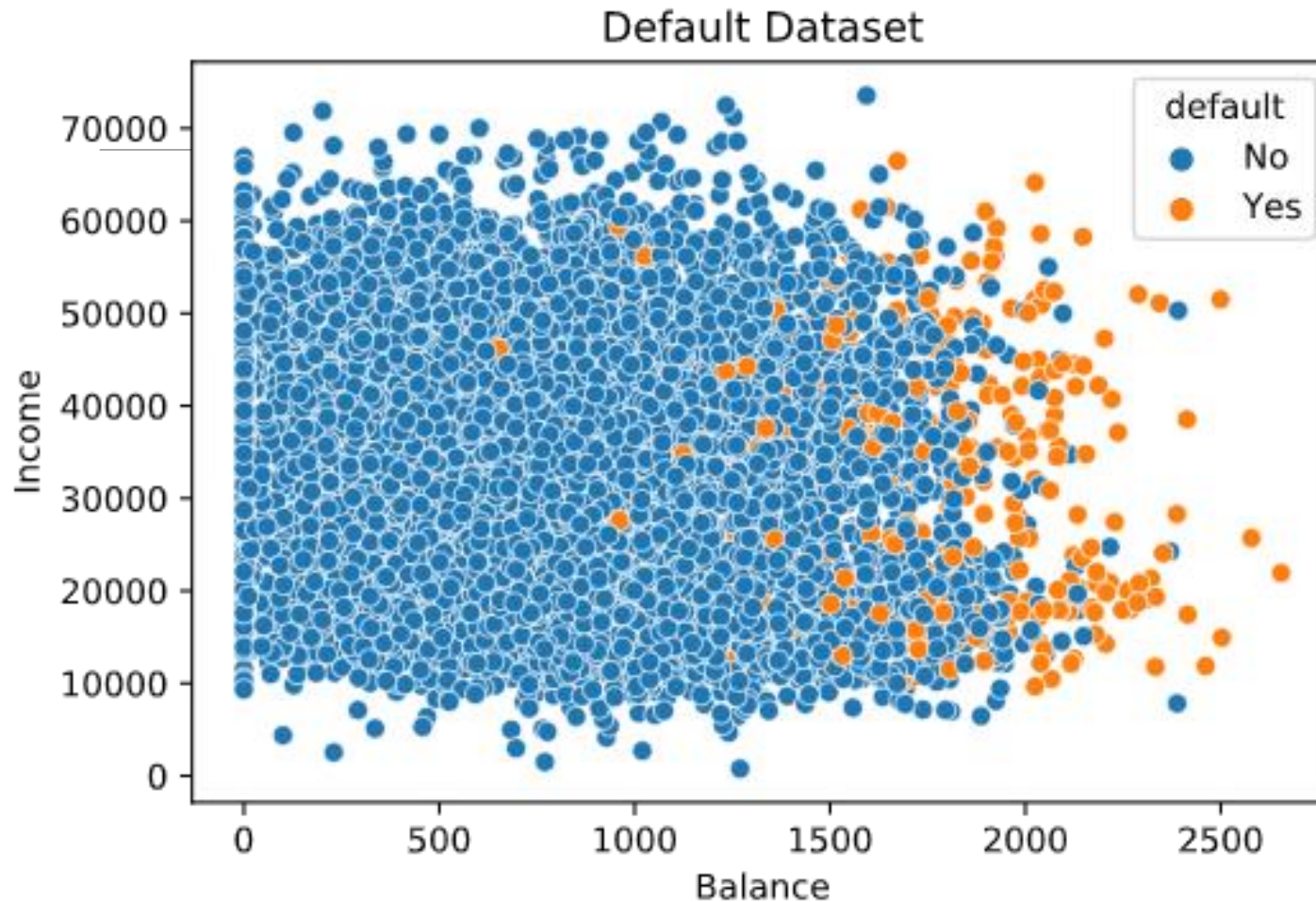
Default dataset: default (yes/no), student (yes/no), balance, income

default	student	balance	income
No	No	729.52	44361.62
No	Yes	817.18	12106.13
No	No	1073.54	31767.13
No	No	529.25	35704.49
No	No	785.65	38463.49

Linear models of classification

1	Balance only			
	coefficient	std error	z	P>|z|
const	-10.6513	0.361	-29.491	0
balance	**0.0055**	0	24.952	0

3	Student only			
	coefficient	std error	z	P>|z|
const	-3.5041	0.071	-49.554	0
student	**0.4049**	0.115	3.52	0

2	Income only			
	coefficient	std error	z	P>|z|
const	-3.0941	0.146	-21.156	0
income	**-8.35E-06**	4.21E-06	-1.985	0.047

4	Balance, income, student			
	coefficient	std error	z	P>|z|
const	-10.869	0.492	-22.079	0
balance	**0.0057**	0	24.737	0
income	**3.03E-06**	8.20E-06	0.37	0.712
Student [yes]	**-0.6468**	0.236	-2.738	0.006

coefficient is statistically significant when p-value is small

From 1, 2 and 3, balance, student and income are significant. Is this correct?

From considering all independent variables (predictors) shows different results.

> Performing regressions with just **one predictor** can be misleading.
> Results may differ significantly from using multiple predictors, specially when predictors are **correlated.**

Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, 2013.

Linear models of classification

Income positive impact on default

Being student negative impact on default

| Individual | balance | 0.0057 | income | **3.03E-06** | Student[yes] | **-0.6468** |
| All | balance | 0.0055 | income | **-8.35E-06** | Student[yes] | **0.4049** |

Income negative impact on default

Being student positive impact on default

A confounder, also known as a confounding variable, extraneous determinant, or lurking variable, is a factor that **affects both the independent and dependent variables**, leading to a misleading connection.

Confounding is a concept related to causality, and it cannot be solely explained through correlations or associations.

student ⟷ balance ⟷ default

Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, 2013.
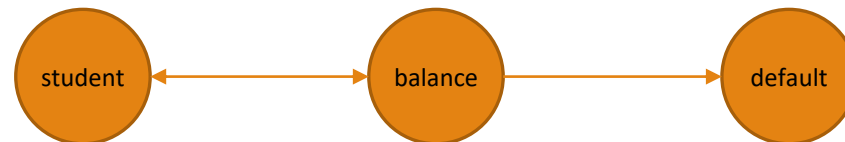
Linear models of classification

Confounding:

➤ Studying and Exam Scores: Hours studied (independent) and exam scores (dependent).

➤ Positive Correlation: More studying linked to higher scores.

➤ Confounding Variable Impact: Factors like natural intelligence or prior knowledge can affect **both studying and scores**.

➤ Example: High intelligence students **study more** and **good** in exams.

➤ Avoiding Misinterpretation: Ignoring confounder may lead to hours studied corresponds solely to exam scores.

Discriminant functions

➢ Discriminant function: a discriminant is a mathematical function that takes an input vector **x** and assigns it to a specific class among a set of K classes.

➢ The discriminant function's task is to compute $p(y = c_k|x)$ for each class c_k. Then, assign the instance to the class based on this probability.

➢ Linear discriminants: Decision surfaces are hyperplanes (use linear combinations of input features to separate different classes).

➢ $y(\mathbf{x}) = w_0 + w_1 x_1 + \cdots, + w_D x_D = w_0 + \widetilde{\boldsymbol{w}}^T \mathbf{x}$

➢ w_0 - Bias and $\widetilde{\boldsymbol{w}}$ - weight vector.

➢ Consider binary classification

 ➢ Decision rule： $y(\mathbf{x}) > 0$, class 1, else class 0

Discriminant functions

Two points $\mathbf{x_A}$, $\mathbf{x_B}$ which are in line ➔ $w_0 + \widetilde{\boldsymbol{w}}^T \mathbf{x_A} = w_0 + \widetilde{\boldsymbol{w}}^T \mathbf{x_B}$

$\widetilde{\boldsymbol{w}}^T (\mathbf{x_A} - \mathbf{x_B})=0$ ➔ vector $\widetilde{\boldsymbol{w}}$ is orthogonal to every vector lying within the decision surface

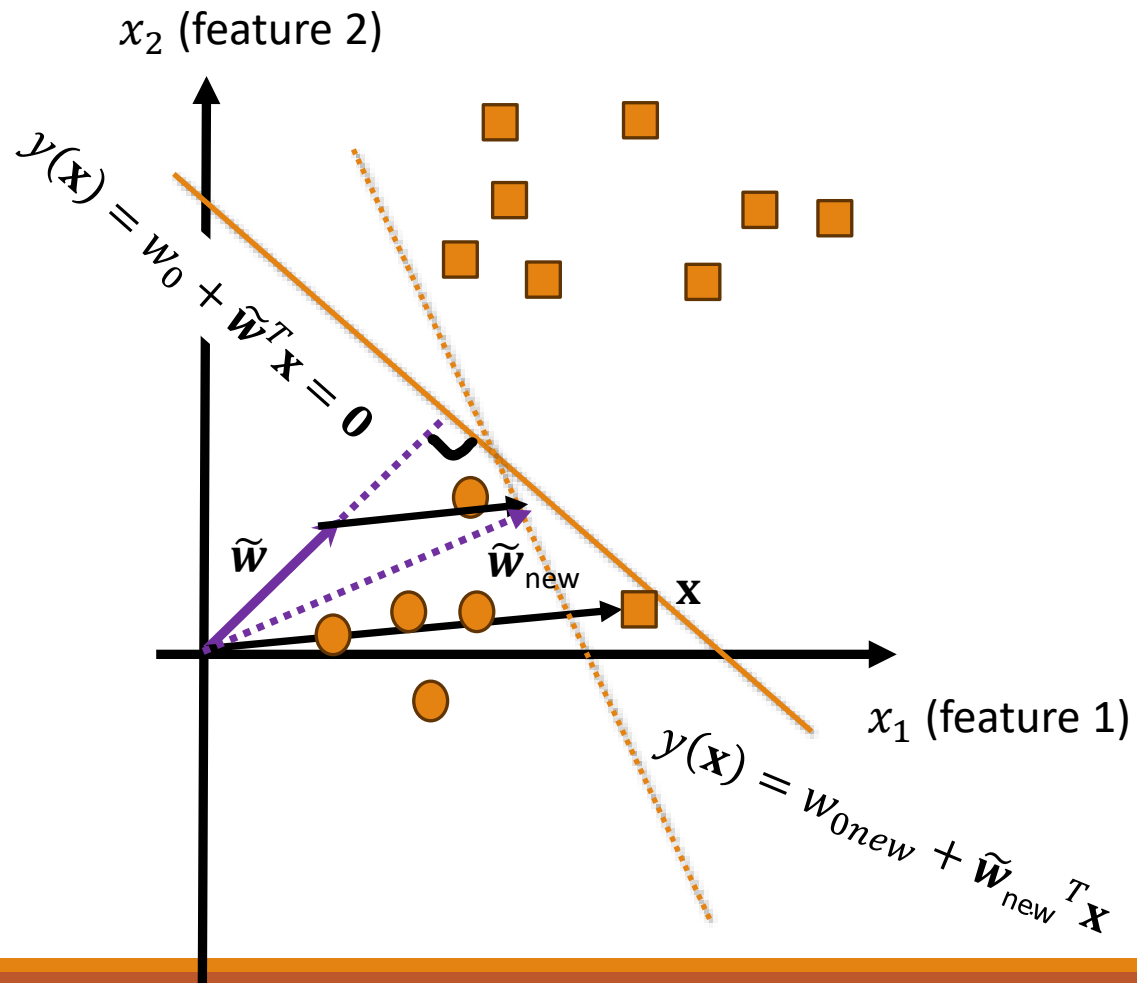$\widetilde{\boldsymbol{w}}$ determines the orientation of the surface.

Distance to the surface from origin is $\dfrac{\widetilde{\boldsymbol{w}}^T \mathbf{x}}{\|\widetilde{\boldsymbol{w}}\|} = \dfrac{-w_0}{\|\widetilde{\boldsymbol{w}}\|}$.

w_0 determines the location of the decision surface

Distance from any vector ($\mathbf{x}$) to the surface is $\dfrac{y(\mathbf{x})}{\|\widetilde{\boldsymbol{w}}\|}$

x_2(feature 2)

y(**x**)> 0

$y(\mathbf{x}) = w_0 + \widetilde{\boldsymbol{w}}^T \mathbf{x} = 0$

x

$\dfrac{y(\mathbf{x})}{\|\widetilde{\boldsymbol{w}}\|}$

$\widetilde{\boldsymbol{w}}$

$y(\mathbf{x})< 0$

x_1 (feature 1)

$\dfrac{\widetilde{\boldsymbol{w}}^T \mathbf{x}}{\|\widetilde{\boldsymbol{w}}\|} = \dfrac{-w_0}{\|\widetilde{\boldsymbol{w}}\|}$

Discriminant functions

x_2 (feature 2)

$y(\mathbf{x}) = w_0 + \widetilde{\mathbf{w}}^T \mathbf{x} = \mathbf{0}$

$\widetilde{\mathbf{w}}$

$\widetilde{\mathbf{w}}_{new}$

$\mathbf{x}$

x_1 (feature 1)

$y(\mathbf{x}) = w_{0new} + \widetilde{\mathbf{w}}_{new}{}^T \mathbf{x}$
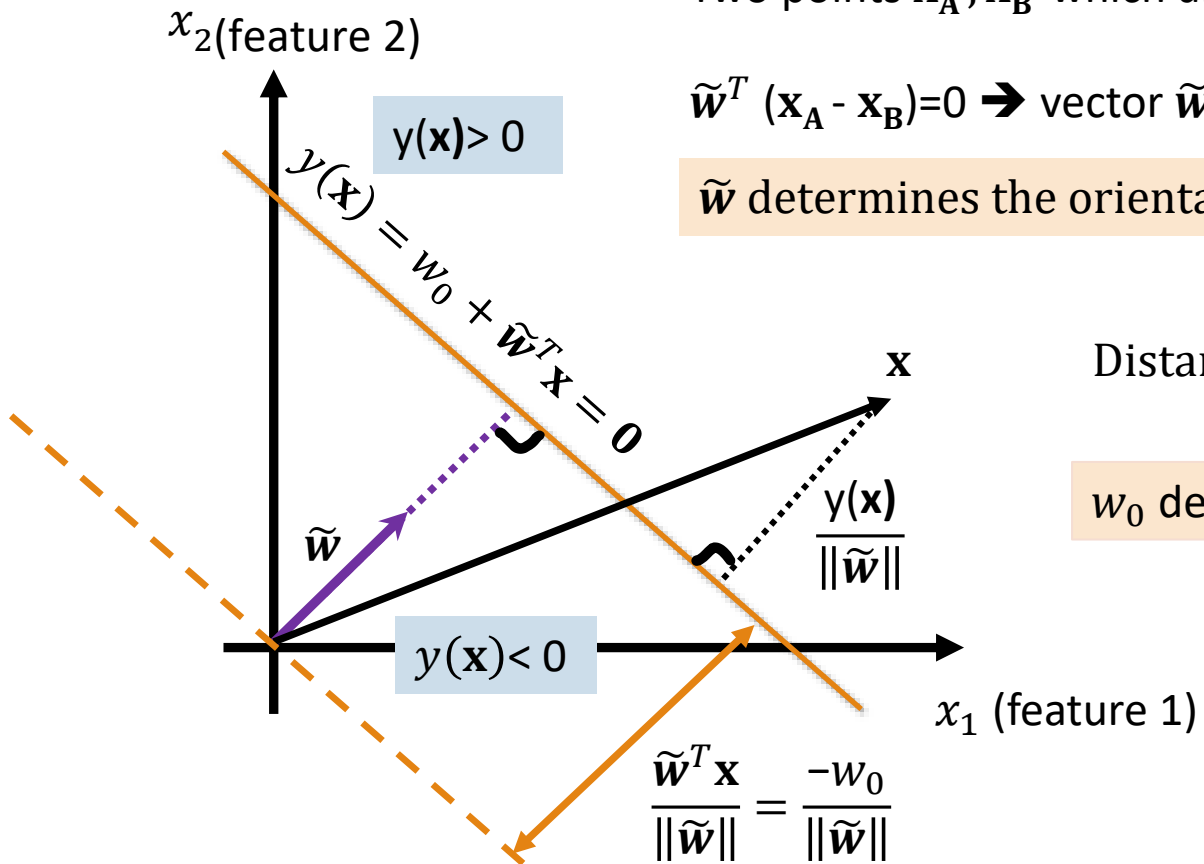
$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \left(\text{sigm}(\mathbf{w}^T \mathbf{x}_i) - y_i \right) \mathbf{x}_i$$

Discriminant functions

➤ Extension to multiple classes
  ➤ One-versus-one : Train many classifiers as there are pairs of labels
  ➤ One-versus-rest : Each class as an independent class and consider the rest combined as only one class

ambiguous regions

One-versus-rest

One-versus-one
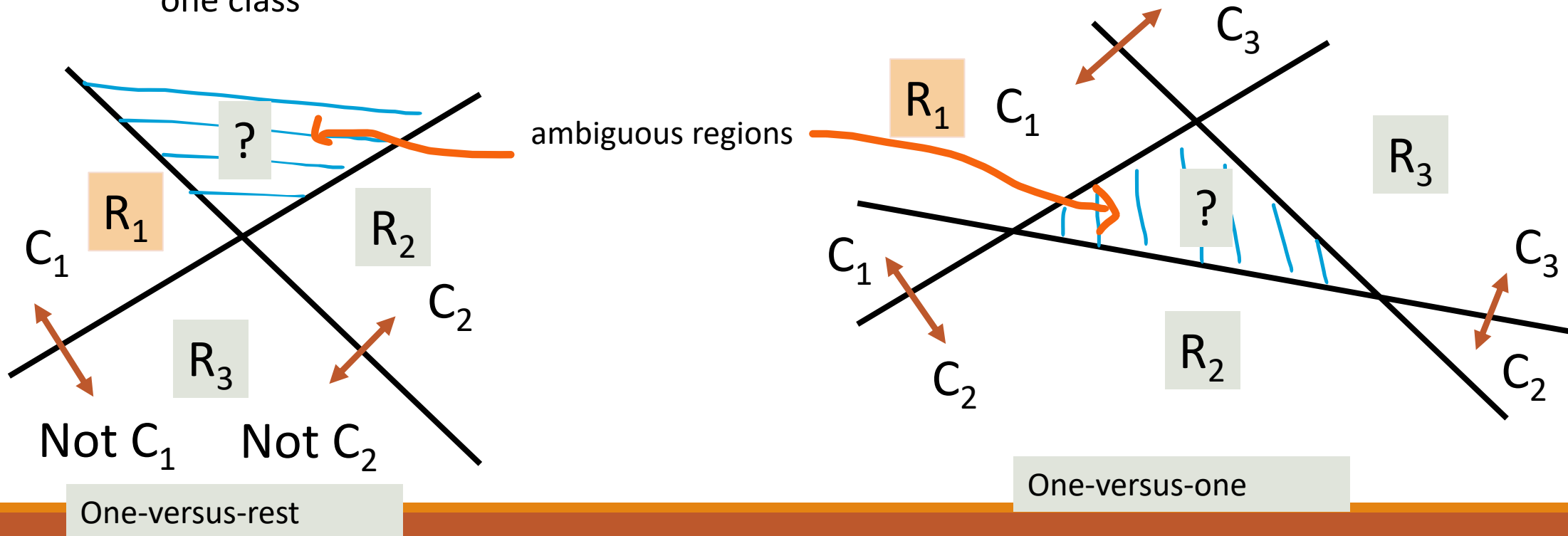
Discriminant functions

➢ Single K-class discriminant : $y_k(\mathbf{x}) = w_{0k} + \widetilde{\boldsymbol{w}}_k{}^T \mathbf{x} = \boldsymbol{w}_k{}^T \mathbf{x}$

➢ Decision boundary between class i and j are given by

➢ $y_i(\mathbf{x}) = y_j(\mathbf{x}) = 0$

➢ Hyperplane is given by $(\boldsymbol{w}_i - \boldsymbol{w}_j)^T \mathbf{x} + (w_{0i} - w_{0j}) = 0$

Aims to directly classify instances into multiple classes using a decision boundary with K linear functions, each associated with a specific class. This is known as "multinomial".

R_i

R_j

R_k
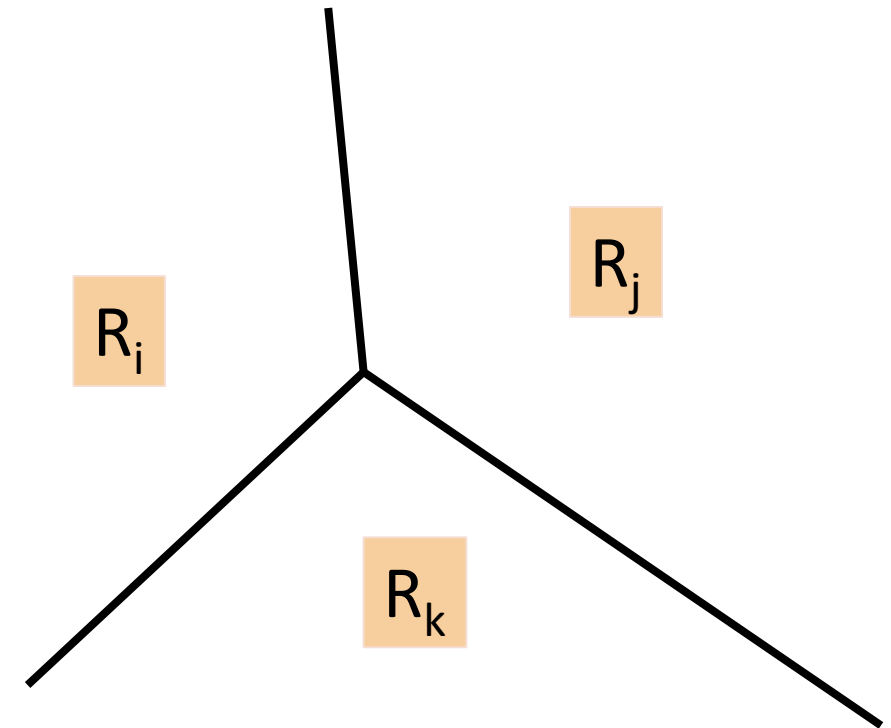
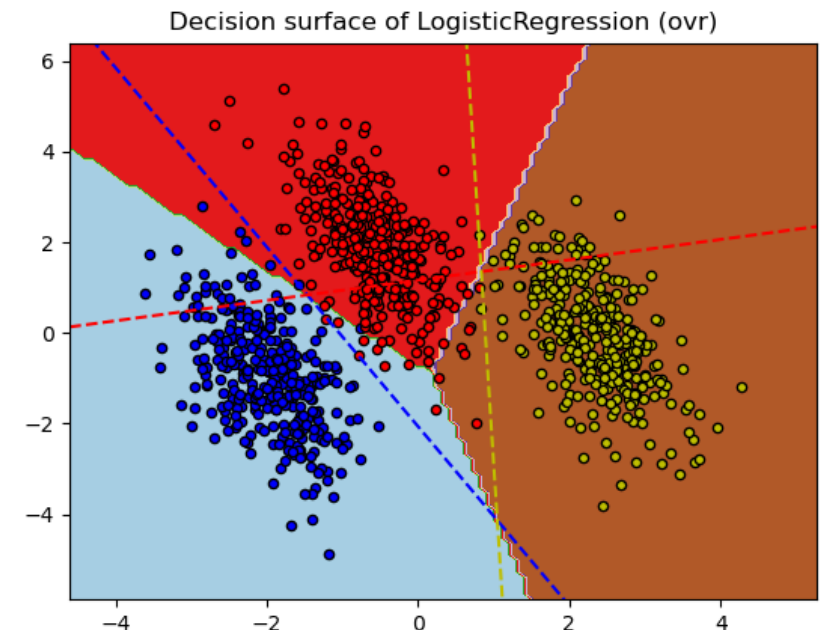# One-vs-the-rest (OvR) multiclass (multinomial)

➢There are methods other than logistic regression that tend to perform better for multiclass classification. This is because logistic regression is inherently a binary classification algorithm.

➢Multinomial classifier directly models the relationships between all classes simultaneously.

➢The One-vs-Rest (OvR). It breaks down the multiclass problem into multiple binary classification sub-problems. For each class, a separate binary classifier is constructed.

➢ Multinomial
Accuracy: 0.995 Precision: 0.99 Recall: 0.995 F1-Score: 0.99
➢ OvR
Accuracy: 0.976 Precision: 0.97 Recall: 0.976 F1-Score: 0.97

Decision surface of LogisticRegression (multinomial)

Decision surface of LogisticRegression (ovr)

One-vs-the-rest (OvR) multiclass

➢Multinomial Logistic Regression provides improved accuracy and faster training on larger datasets.

Multinomial vs One-vs-Rest Logistic L1
Dataset 20newsgroups

Model: One versus Rest
Model: Multinomial

Multiclass sparse logistic regression on 20newgroups — scikit-learn 1.3.0 documentation
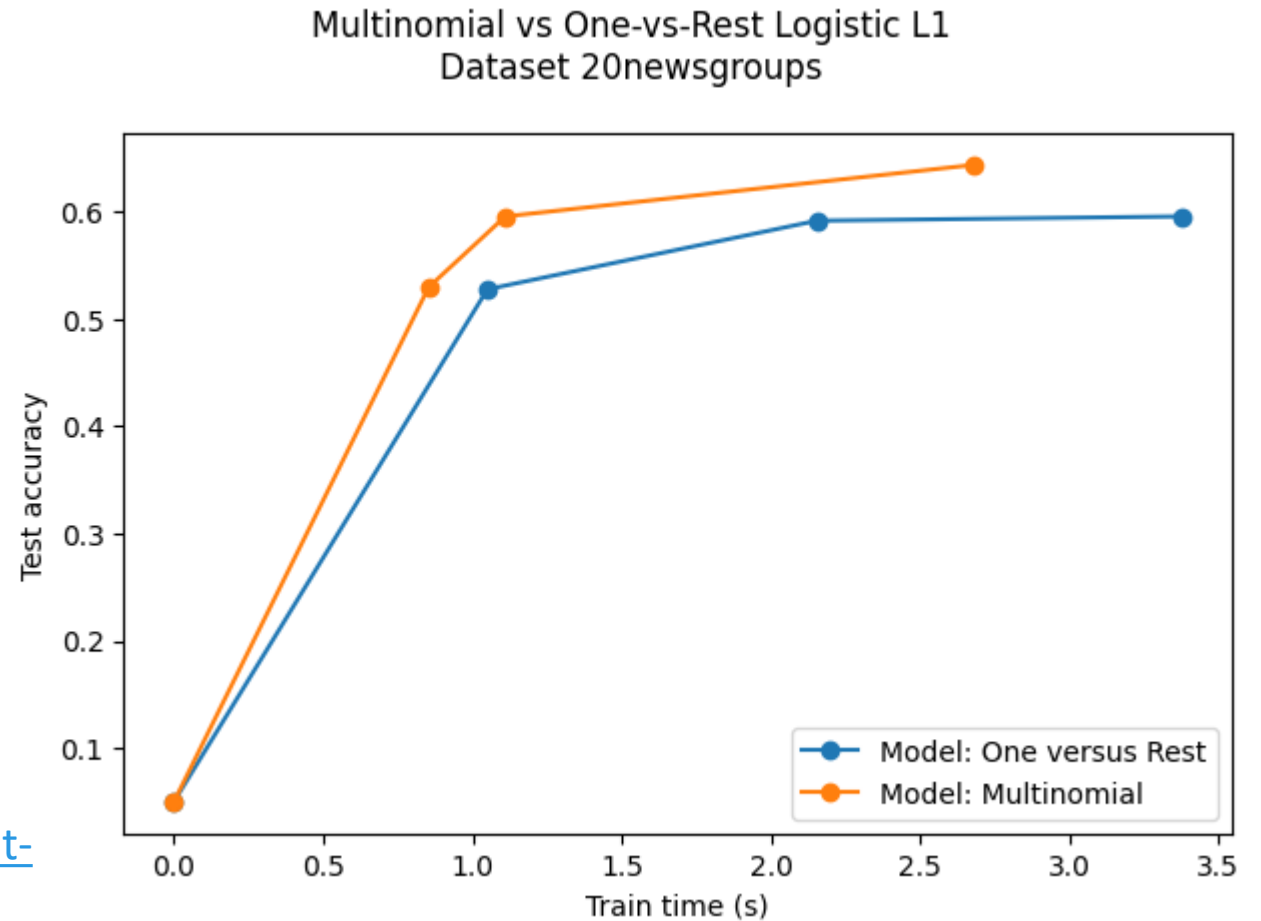
Classification example

➢ Logistic regression in iris dataset

➢ Two features are used

➢ Accuracy: 0.9 Precision: 0.90

➢ Recall: 0.9 F1-Score: 0.89

Decision surface of Logistic Regression

Confusion Matrix (Percentages)
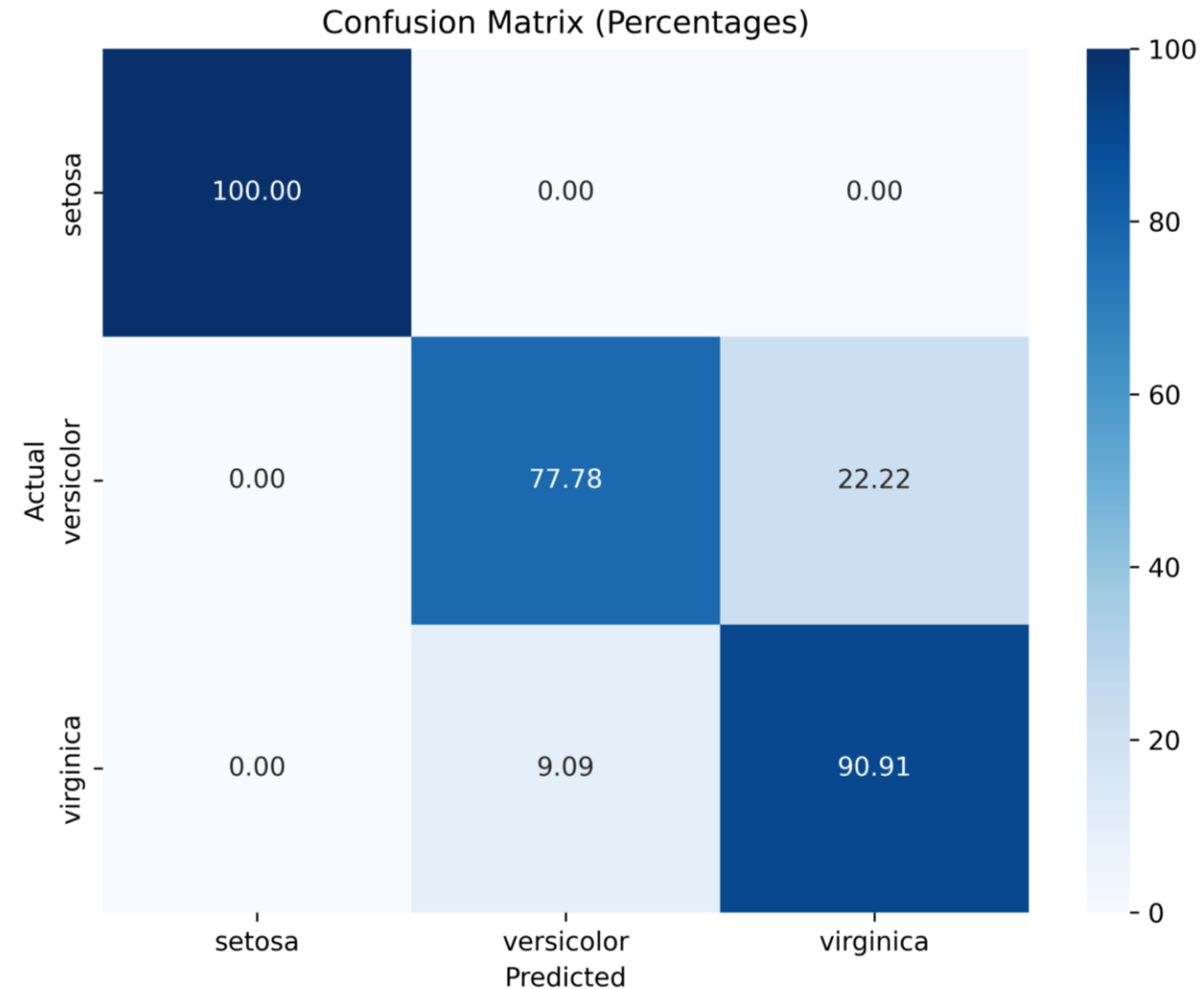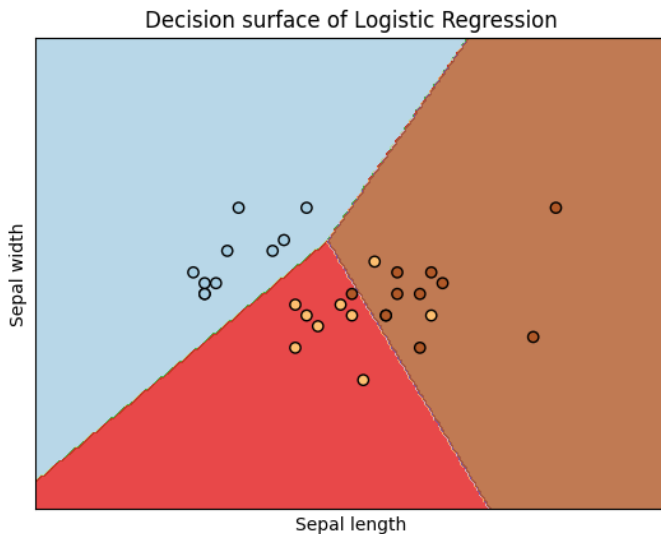
Classification example

➢Logistic regression in iris dataset

➢Using all features

➢sepal length, sepal width,

petal length, petal width

Confusion Matrix (Percentages)
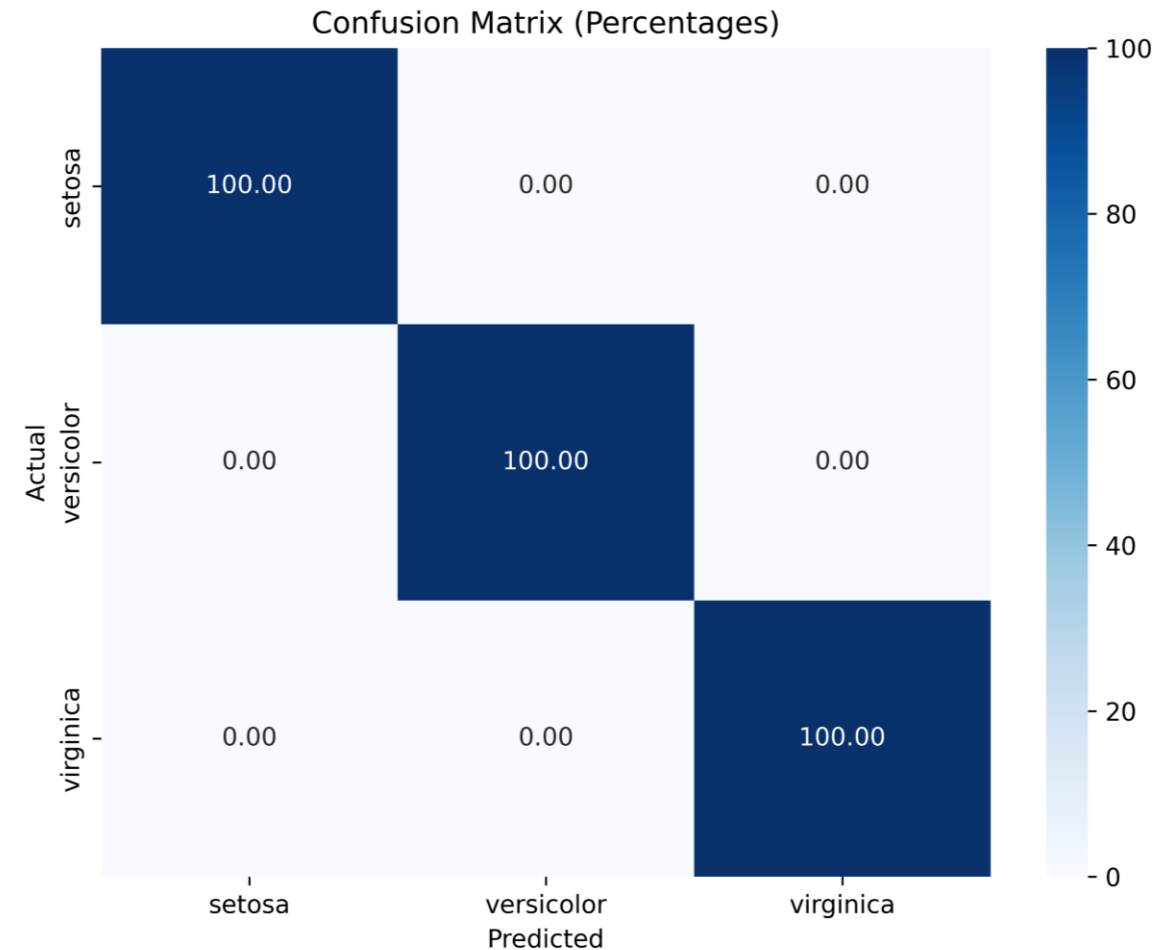
Classification example

➤ Logistic Regression with L1- penalty

➤ Digits 0 to 4 class 0 and 5 to 9 class 1
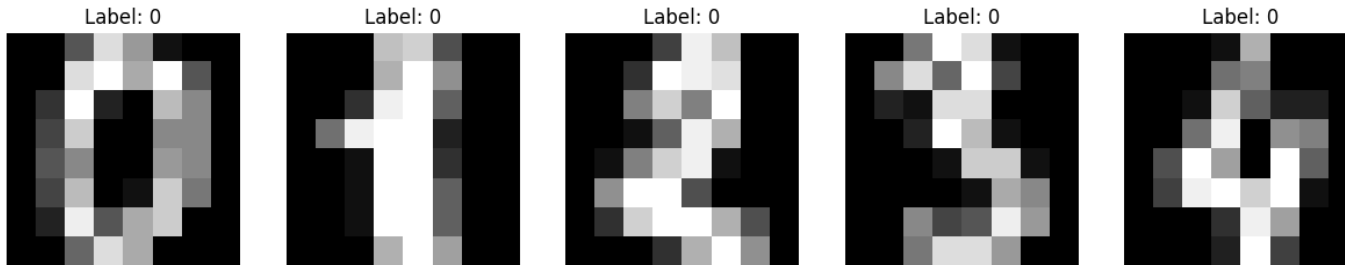
|	C=1.00	C=0.10	C=0.01
Sparsity with L1 penalty:	4.69%	29.69%	84.38%
Sparsity with Elastic-Net penalty:	4.69%	14.06%	68.75%
Sparsity with L2 penalty:	4.69%	4.69%	4.69%
Score with L1 penalty:	0.90	0.90	0.86
Score with Elastic-Net penalty:	0.90	0.90	0.88
Score with L2 penalty:	0.90	0.90	0.89
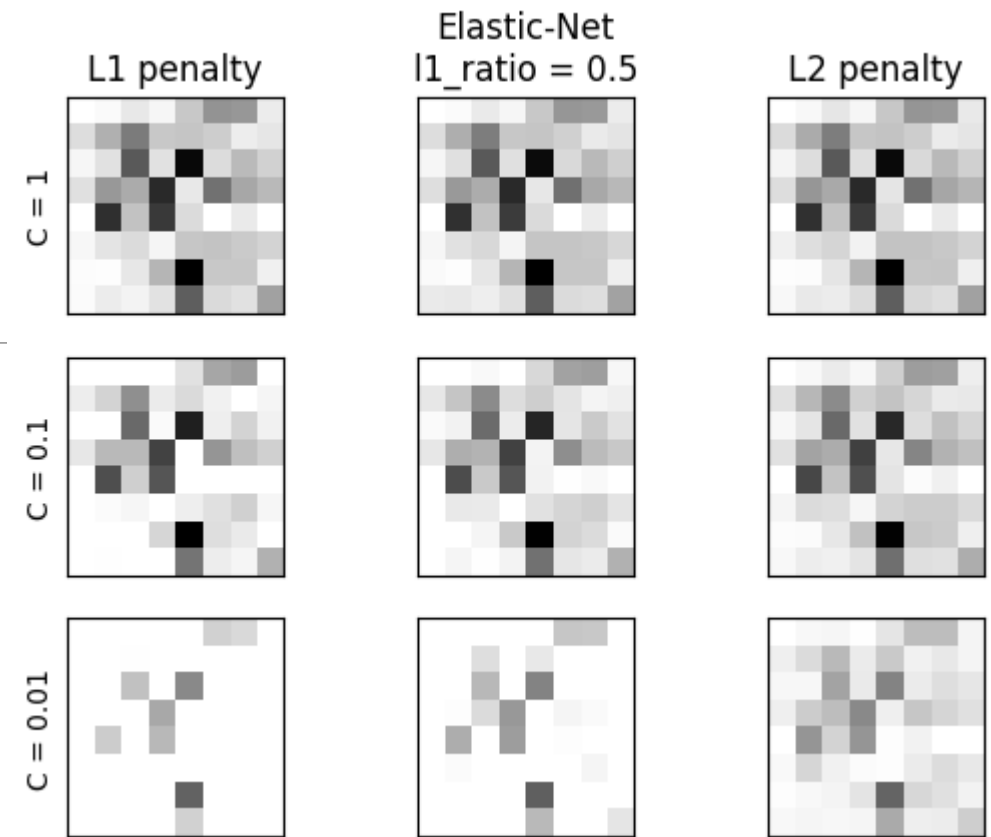
➤ Larger C values provide the model more freedom to fit the data.
➤ With L1 penalty and smaller C, the model tends to have fewer significant features, leading to sparser solutions.
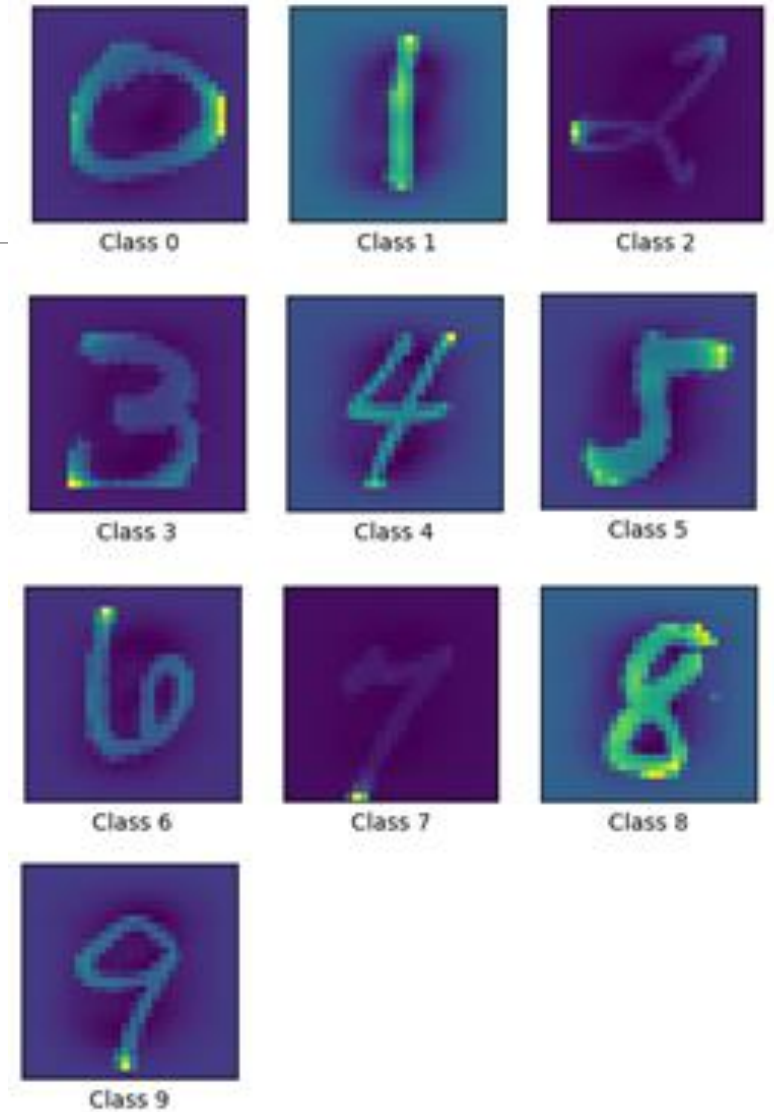➤ Elastic-Net penalty's sparsity is between that of L1 and L2 penalties.

Classification example

➤ MNIST classification using multinomial logistic + L1 penalty
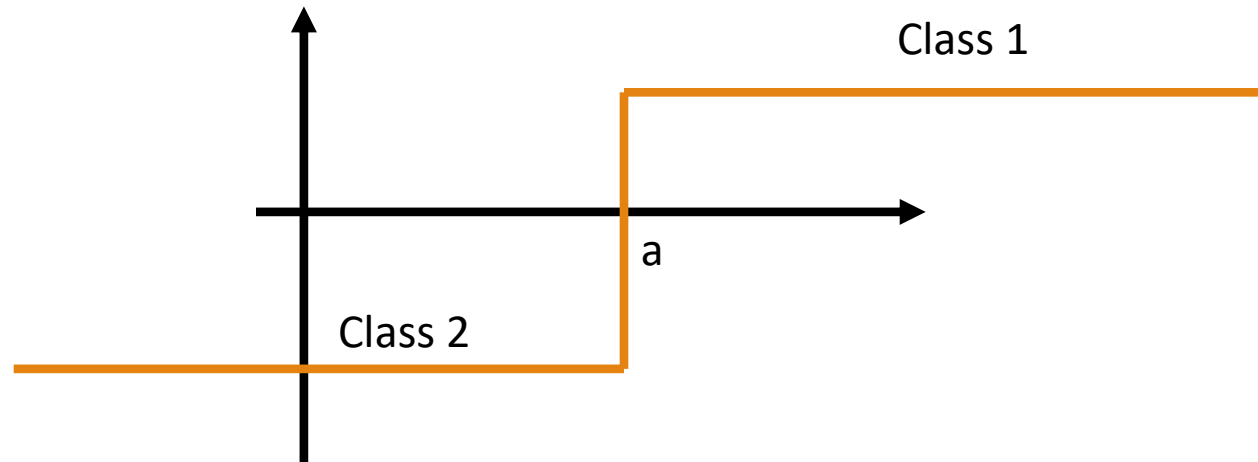
Weights for each class
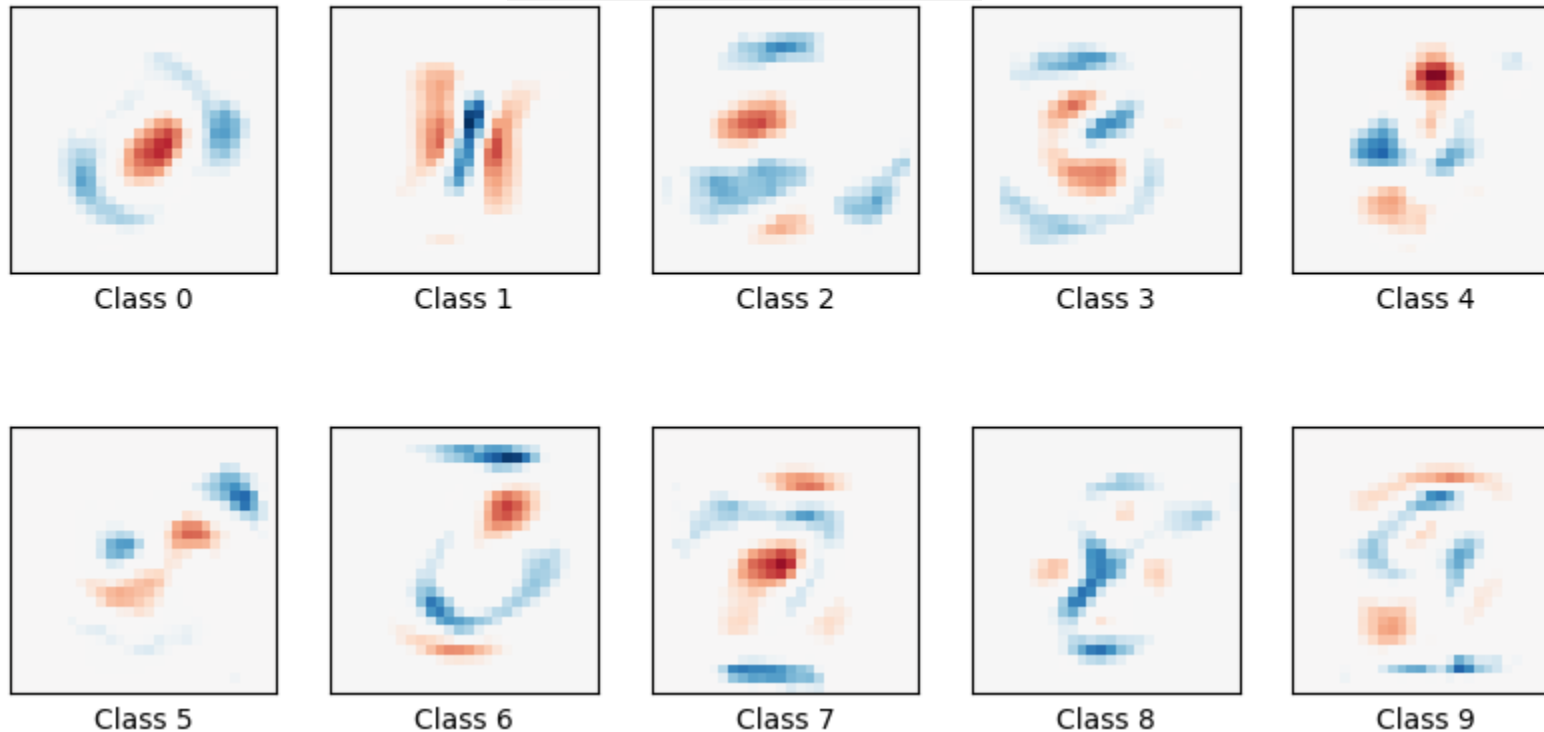
Example digit for each class

Perceptron

➢ The perceptron was introduced by Frank Rosenblatt as a simple binary classification algorithm.

➢ $p(y_i = 1|x_i, \mathbf{w}) = f(\mathbf{w}^T\mathbf{x})$. Here, f() is nonlinear activation function.

$$f(a) = \begin{cases} +1, & \text{if } a \geq 0 \\ -1, & \text{if } a < 0 \end{cases}$$

Class 1

Class 2

a

Total number of misclassified patterns as the error function leads to several challenges:
➢ The error is constant if the decision boundary doesn't cross data points. When it does cross due to a change in **w**, the error instantly jumps, creating a non-smooth, discontinuous function. Difficult to use simple gradient-based approaches.

Perceptron

➤ Perceptron criterion

➤ Try to find $\mathbf{w}$ that
  ➤ $(\mathbf{w}^T\mathbf{x}) > 0$ for class 1 (represents by c=1)
  ➤ $(\mathbf{w}^T\mathbf{x}) < 0$ for class 0 (represents by c= -1)

  ➤ We try to find $\mathbf{w}$ for data samples y_i $(\mathbf{w}^T\boldsymbol{x}_i) > 0$

➤ The perceptron criterion associates zero error with any pattern that is correctly classified, whereas for a misclassified pattern $\boldsymbol{x}_i$, the following loss function is minimized.

$$E(\boldsymbol{w}) = \sum_{i \in M} -\boldsymbol{w}^T\boldsymbol{x}_i y_i$$

➤ In this formula, the summation is taken over all misclassified patterns.
➤ Gradient descent update

$$\boldsymbol{w}_{new} \leftarrow \boldsymbol{w}_{old} + \alpha\, \boldsymbol{x}_i y_i$$

Limitations of perceptron algorithm
➤ No probabilistic outputs
➤ Not easily adaptable for more than two classes

C. M. Bishop, Pattern Recognition and Machine Learning, Springer NY, 2006

Thank You
Q & A