CS 480/680
Introduction to Machine Learning

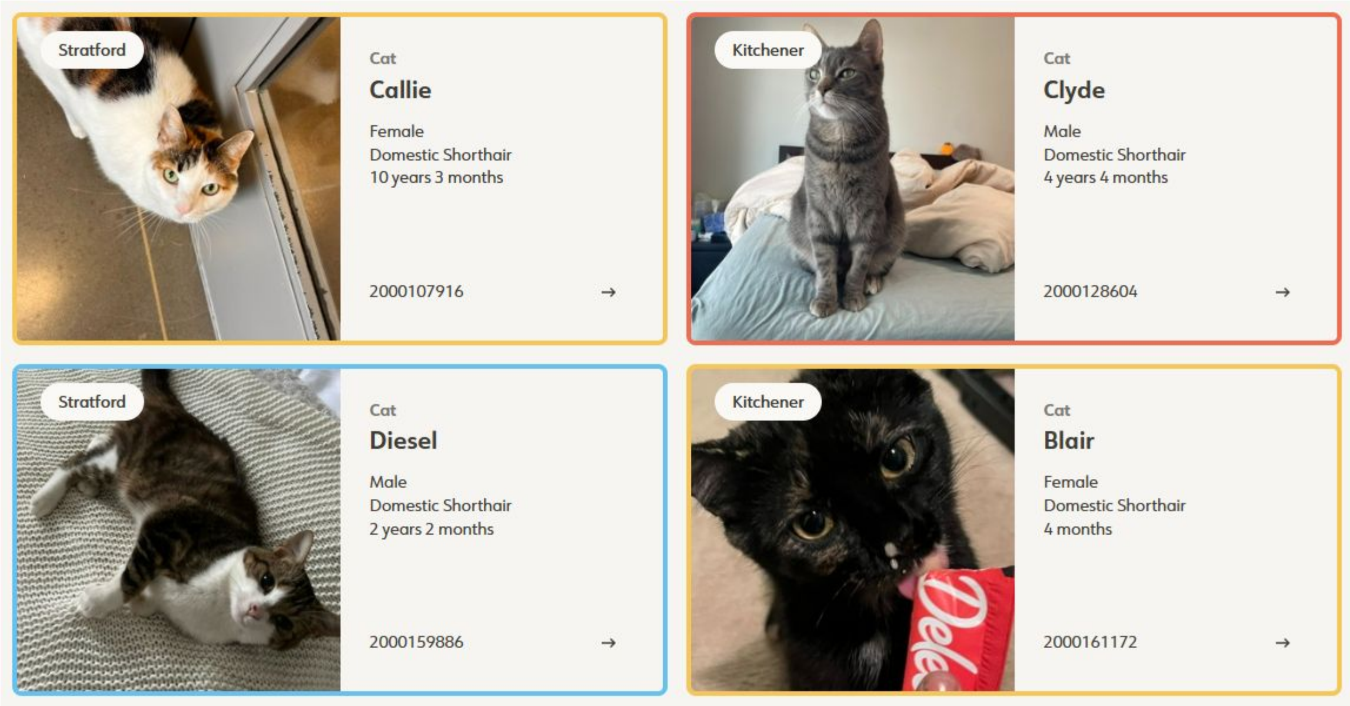
Lecture 5
Logistic Regression and Numerical Optimization

Kathryn Simone

24 September 2024

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Will a shelter cat get adopted within the next 30 days?



Source: Humane Society of Kitchener Waterloo Stratford Perth (Accessed 21/09/2024)

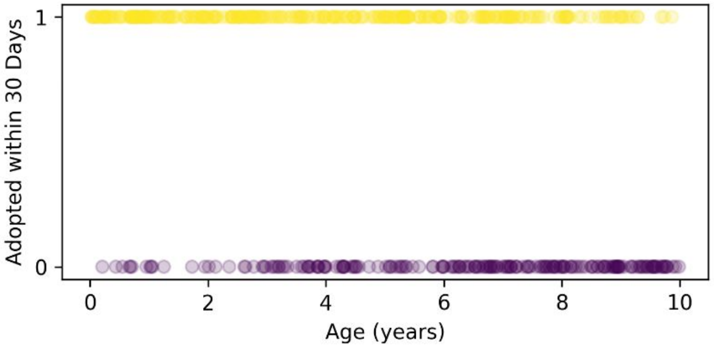
The cat adoption dataset

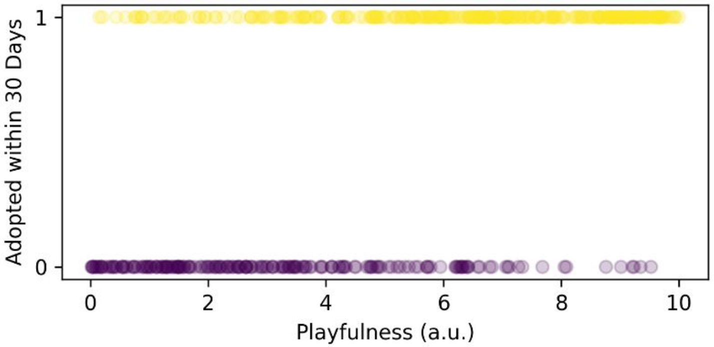
Attributes → **Outcome/Label**

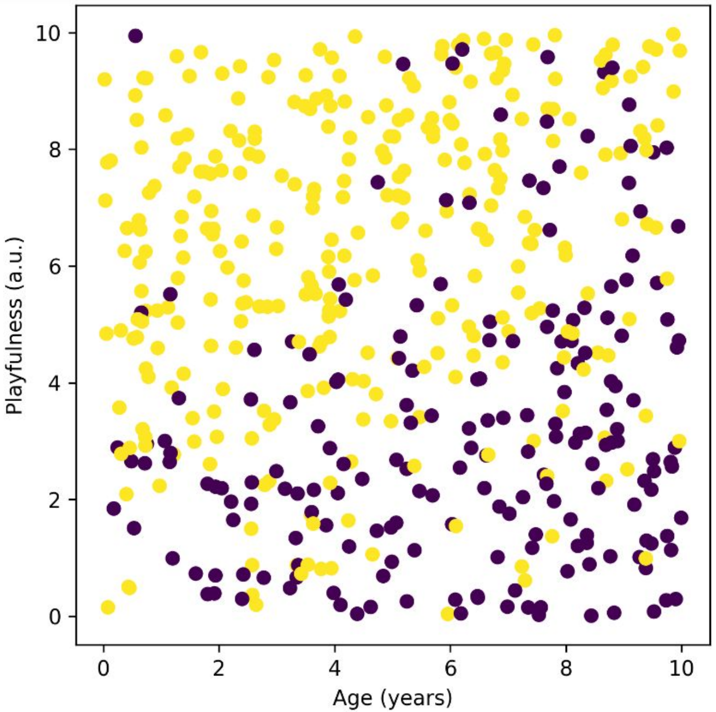
Cats →

Age (Years)	Playfulness (a.u.)	Adopted?
0.3	5	Yes
6	1	No
1	9	Yes
9	7	Yes
0.2	3	Yes

Exploring the cat adoption dataset







Knowledge of the chances of an event guides decision-making

Consider and compare:

Prediction A:
A cat will not get adopted within 30 days.

- Model has binary output
- Classification task

Prediction B:
*The **probability** that a cat will get adopted within 30 days is **5%**.*

- Model has continuous output
- Regression task used for classification
- Can **prioritize** efforts (marketing campaigns, waived/adjusted fees, etc) and **justify** decisions

Key Questions

I. **What is logistic regression?**

II. **How do we estimate the parameters?**

III. **How can we handle the multiclass case?**

Key Questions

I. **What is logistic regression?**

II. How do we estimate the parameters?

III. How can we handle the multiclass case?

How to model the probability of an outcome?

In Linear regression, we
assumed a hypothesis class of the form:

$$p(X) = \beta_0 + \beta_1 X.$$

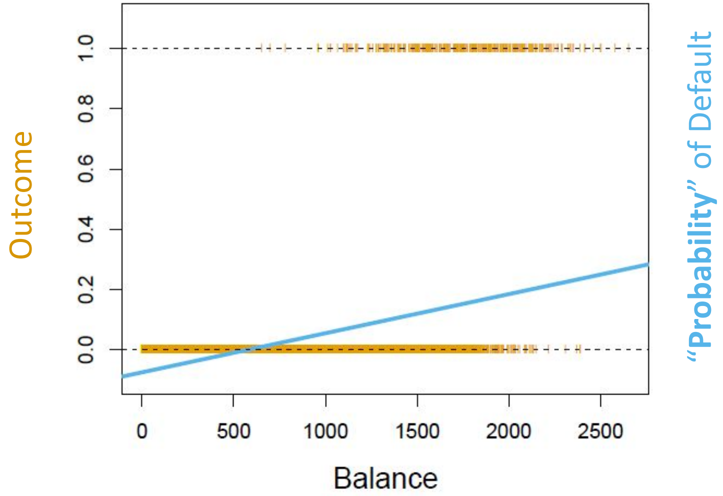


Figure adapted from Introduction to Statistical Learning, Section 4.3

Hypothesis class for logistic regression

Goal: Learn a function $h : \mathbb{R}^d \rightarrow [0, 1]$
Hypothesis class:

$$\mathcal{H} = \left\{ x \rightarrow \frac{1}{1 + e^{-\langle w, x \rangle}} \right\}$$

Where:
$w \in \mathbb{R}^d$ is the parameter vector,
x is the feature vector,
$\phi(z) = \frac{1}{1+e^{-z}}$ is the logistic function.
$h(x)$ can be interpreted as the probability he label associated with a feature vector x is 1.

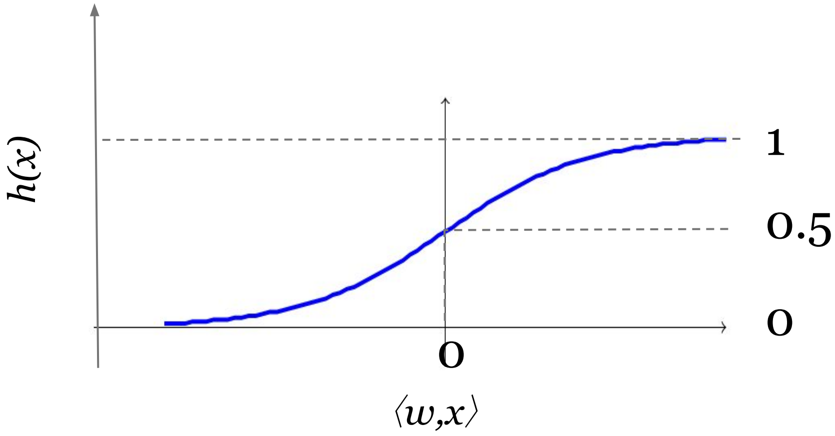
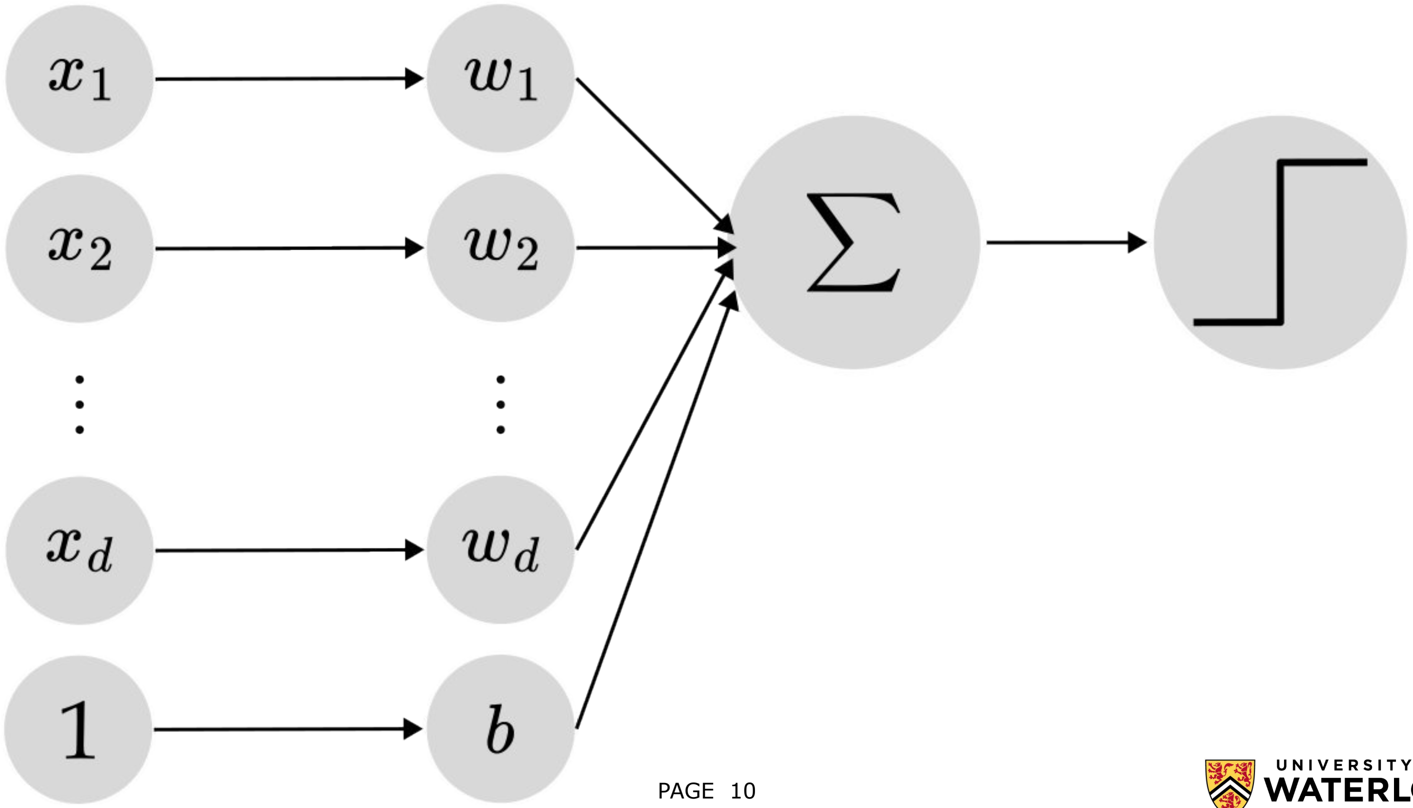
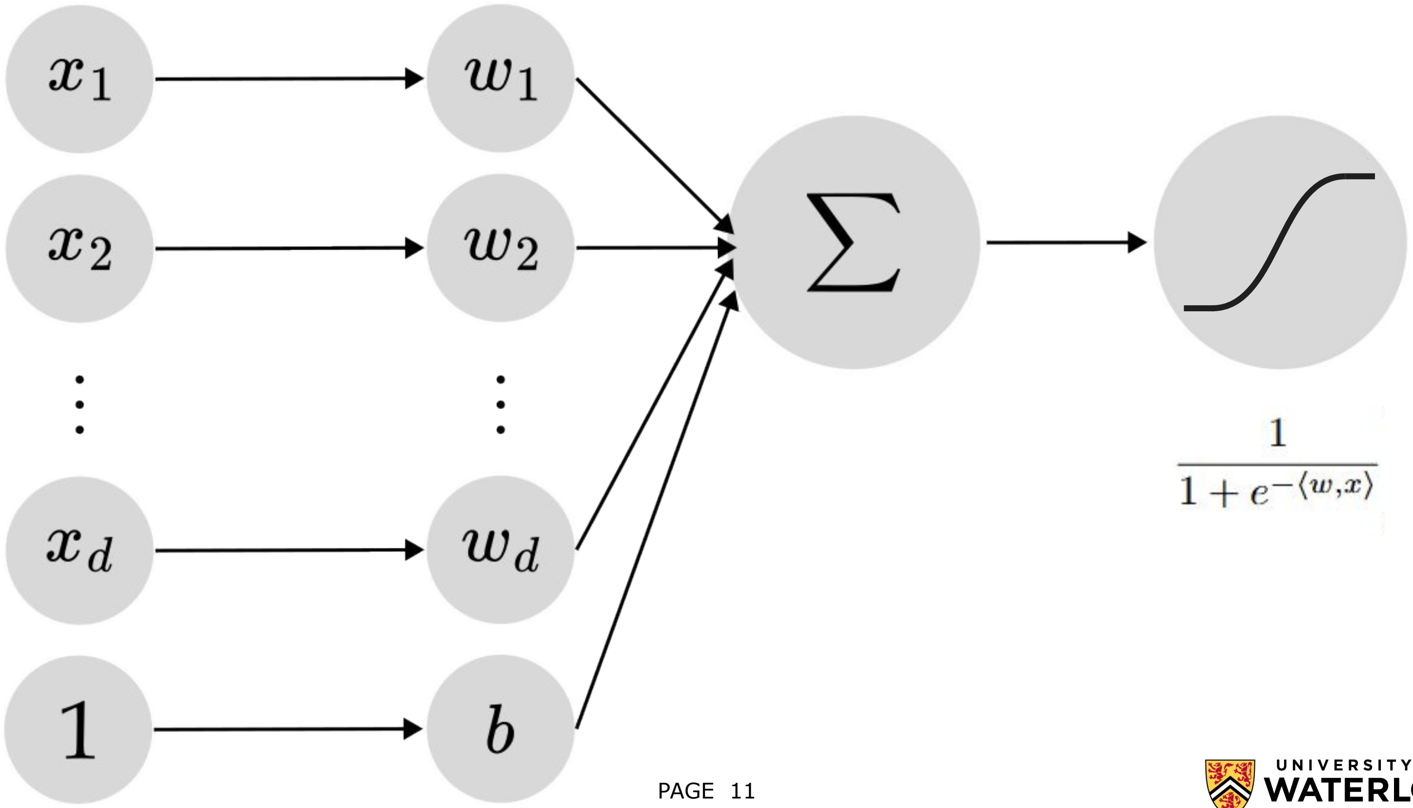


Figure adapted from Understanding Machine Learning, Section 9.3

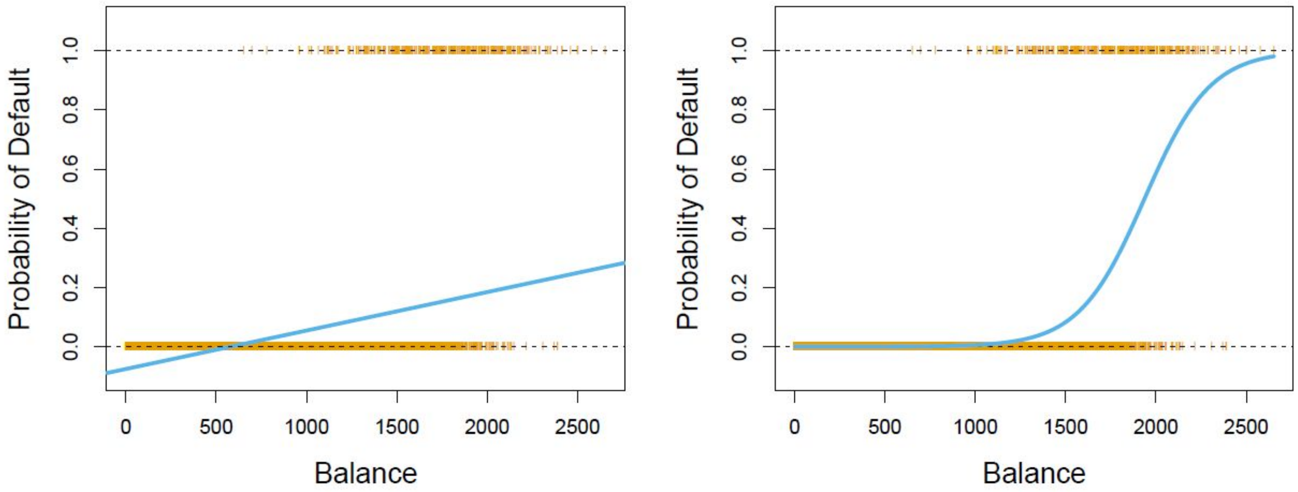
Recall: Perceptron and the class of halfspaces



Compare to Perceptron and class of halfspaces



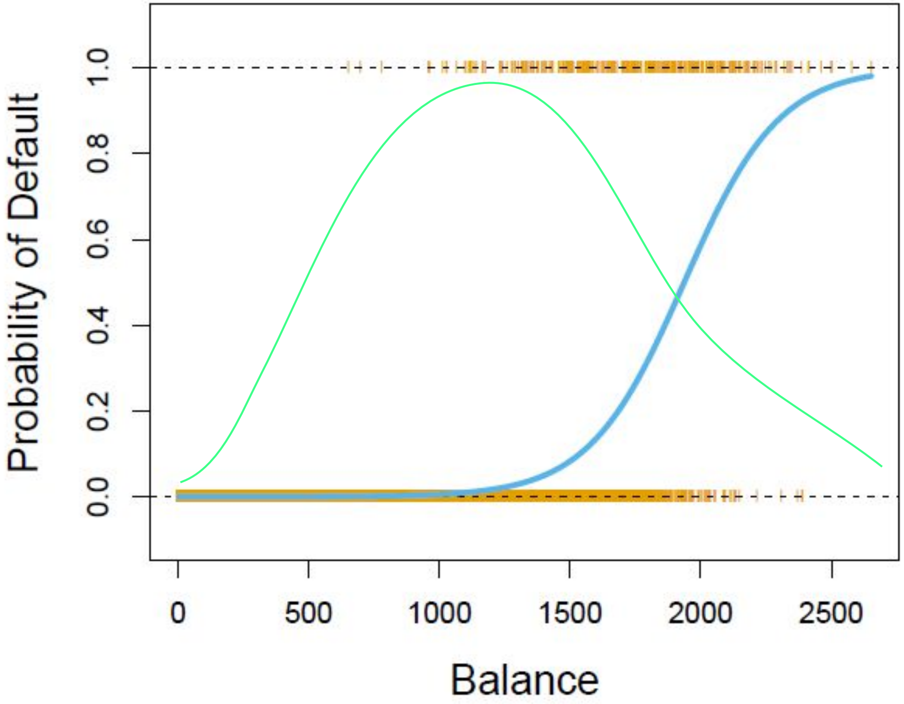
The logistic model for probability of an outcome



$$p(X) = \beta_0 + \beta_1 X.$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

Monotonicity contributes to interpretability



Discussion: Logistic Regression in the Credit Industry
(2nd Order Solutions on medium.com)

Key Questions

I. What is logistic regression?

II. **How do we estimate the parameters?**

III. How can we handle the multiclass case?

Interpreting $h(x)$ as a probability requires a stochastic model of the outcome

$$h(x) = \frac{1}{1 + e^{-\langle w, x \rangle}}$$

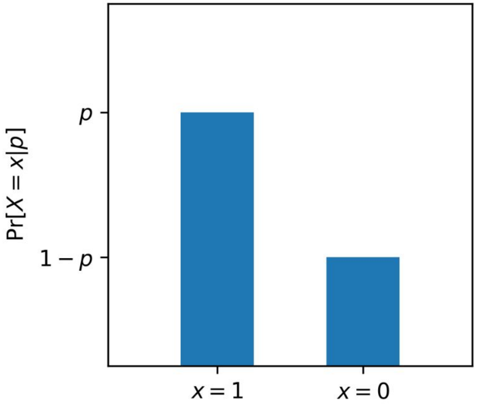
$$\implies h(x) = \Pr[Y = 1 \mid x, w]$$

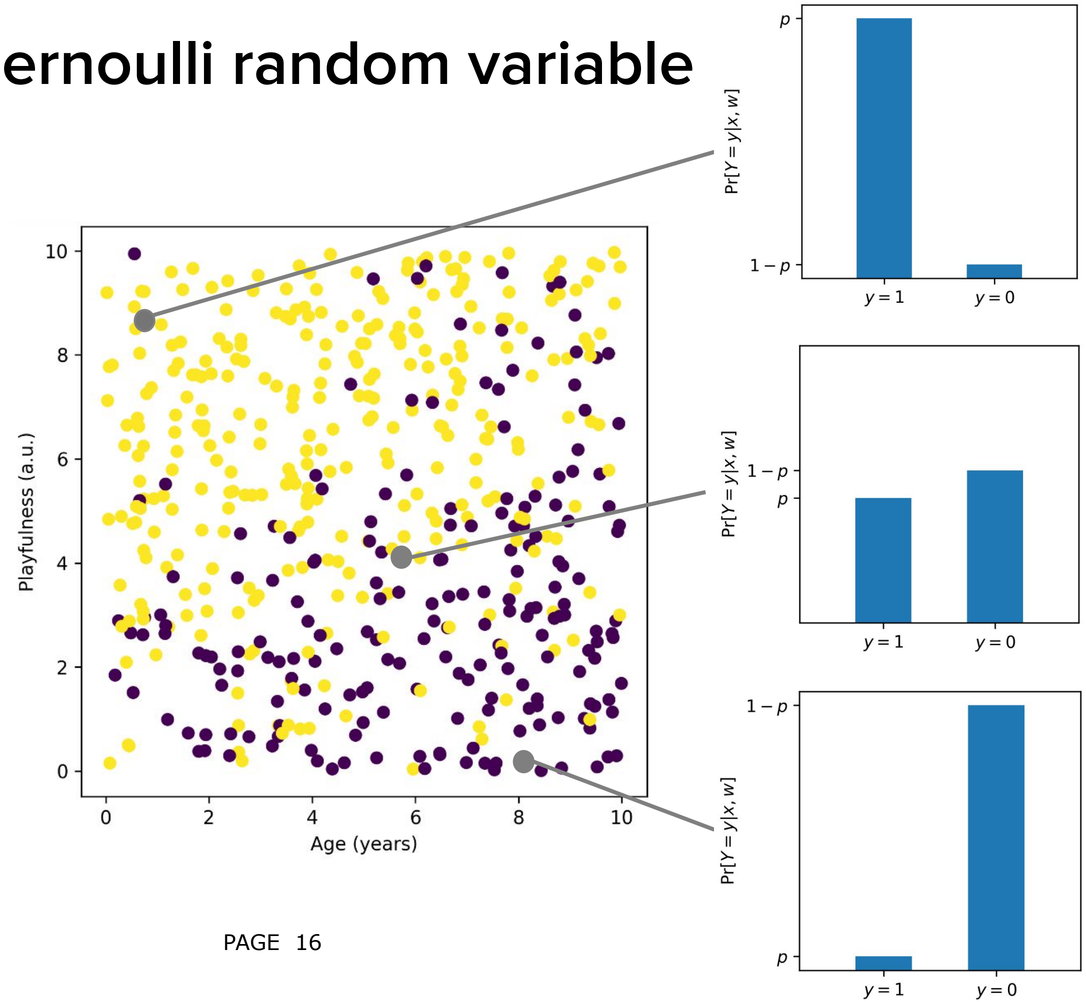
Recall and apply the Bernoulli random variable

Bernoulli random variable:

$$\Pr[X = x] = \begin{cases} p & \text{if } x = 1, \\ 1 - p & \text{if } x = 0, \end{cases}$$

where $0 \leq p \leq 1$.





Deriving the likelihood function starting with the Bernoulli RV

We model the outcome y as a Bernoulli random variable. The likelihood function is defined as:

$$\mathcal{L}(p \mid \boldsymbol{y}) = \prod_{i=1}^{n} p^{y_i}(1-p)^{(1-y_i)},$$

Where p is a parameter, and $\boldsymbol{y}$ denotes the set of n individual observations y_i. To gain intuition for this likelihood, consider $y_i \in \{0, 1\}$:

$$\mathcal{L}(p \mid \boldsymbol{y}) = \begin{cases} \prod_{i=1}^{n} p & \text{for } y_i = 1 \\ \prod_{i=1}^{n} (1-p) & \text{for } y_i = 0 \end{cases}$$

Taking the log of both sides, this reduces to:

$$\begin{aligned} \log \mathcal{L}(p \mid \boldsymbol{y}) &= \sum_{i=1}^{n} \log\left(p^{y_i}(1-p)^{(1-y_i)}\right) \\ &= \sum_{i=1}^{n} y_i \log p + (1-y_i)\log(1-p) \end{aligned}$$

$$\log(ab) = \log a + \log a$$

Deriving the log-likelihood function for logistic regression (1/2)

$$\log\mathcal{L}(p \mid \boldsymbol{y}) = \sum_{i=1}^{n} y_i \log p + (1-y_i)\log(1-p)$$

We seek to reparametrize the likelihood for the logistic hypothesis class, which is our model of the probability of an outcome given the features.

$$h(x) = \frac{1}{1+e^{-\langle w,x\rangle}}$$

$$\implies h(x) = \Pr[Y=1 \mid x, w]$$

Let $p(x_i, w)$ denote $\Pr[Y = 1 \mid x_i, w]$, that is, the probability that observation x_i will be labelled positive. Then

$$p(x_i, w) = \frac{1}{1+e^{-\langle w,x_i\rangle}}.$$

$$\log\mathcal{L}(w \mid \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} y_i \log\left(\frac{1}{1+e^{-\langle w,x\rangle}}\right) + (1-y_i)\log\left(1-\frac{1}{1+e^{-\langle w,x\rangle}}\right)$$

If $y_i = 1$:

$$\log\mathcal{L}(w \mid \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \log\left(\frac{1}{1+e^{-\langle w,x_i\rangle}}\right)$$

Using $\log_b \frac{1}{b} = -\log_a b$

$$= \sum_{i=1}^{n} -\log\left(1+e^{-\langle w,x_i\rangle}\right)$$

Similarly, if $y_i = 0$:

$$\log\mathcal{L}(w \mid \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \log\left(1-\frac{1}{1+e^{-\langle w,x_i\rangle}}\right)$$

$$= \sum_{i=1}^{n} -\log(1+e^{\langle w,x_i\rangle})$$

Full derivation at the end of this deck, if interested

Deriving the log-likelihood function for logistic regression (2/2)

$$\log \mathcal{L}(w \mid \boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \sum_{i=1}^{n} -\log\left(1 + e^{-\langle w, x_i \rangle}\right) & \text{for } y_i = 1 \\ \sum_{i=1}^{n} -\log\left(1 + e^{\langle w, x_i \rangle}\right) & \text{for } y_i = 0 \end{cases}$$

If we let

$$\tilde{y_i} = \begin{cases} +1 & \text{for } y_i = 1 \\ -1 & \text{for } y_i = 0, \end{cases}$$

Then we can arrive at a compact expression for the log-likelihood of the paramter vector w

$$\log \mathcal{L}(w \mid \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} -\log\left(1 + e^{\tilde{y_i} \langle w, x_i \rangle}\right)$$

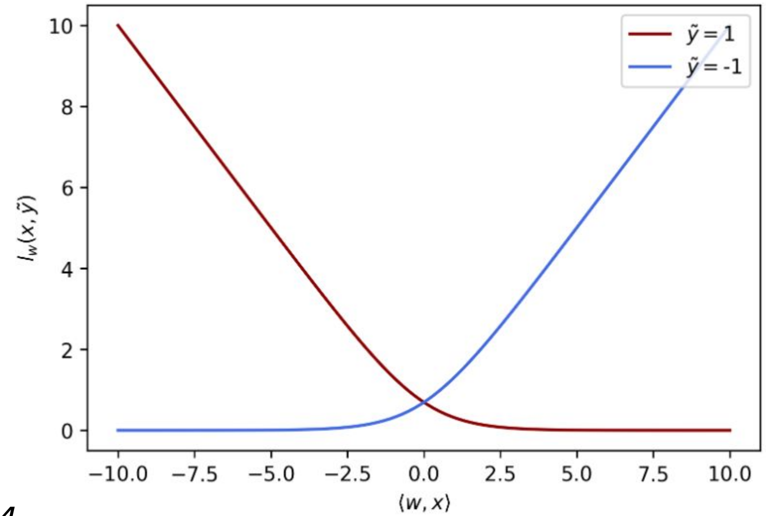
The logistic regression objective and cross-entropy loss

We want to estimate w, that is, find some $\hat{w}$ that maximizes the likelihood of the data:

$$\hat{w} = \text{argmax}_w \sum_{i=1}^{n} -\log\left(1 + e^{-\tilde{y_i}\langle \hat{w}, x_i \rangle}\right)$$
$$= \text{argmin}_w \sum_{i=1}^{n} \log\left(1 + e^{-\tilde{y_i}\langle \hat{w}, x_i \rangle}\right)$$
$$= \text{argmin}_w \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-\tilde{y_i}\langle \hat{w}, x_i \rangle}\right)$$
$$\implies E[l_w(x_i, y_i)] = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-\tilde{y_i}\langle \hat{w}, x_i \rangle}\right)$$
$$\implies l_w(x, \tilde{y}) = \log\left(1 + e^{-\tilde{y}\langle \hat{w}, x \rangle}\right)$$

$$l_w(x, \tilde{y}) = \log\left(1 + e^{-\tilde{y}\langle \hat{w}, x \rangle}\right)$$

	$\langle w, x \rangle \ll 0$	$\langle w, x \rangle \gg 0$
$\tilde{y} = -1$	$l_w(x, \tilde{y}) \approx \log(1)$	$l_w(x, \tilde{y}) \approx \langle \hat{w}, x \rangle$
$\tilde{y} = +1$	$l_w(x, \tilde{y}) \approx \langle \hat{w}, x_i \rangle$	$l_w(x, \tilde{y}) \approx \log(1)$



Proof of convexity: Probabilistic Machine Learning, Section 10.2.3.4

Gradient descent for numerical optimization

Recall that the gradient of a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ at w is denoted $\nabla f(w)$, and is the vector of partial derivatives of f.

In **gradient descent**, the parameter vector w is updated in the direction opposite to that of the gradient, with step size η:

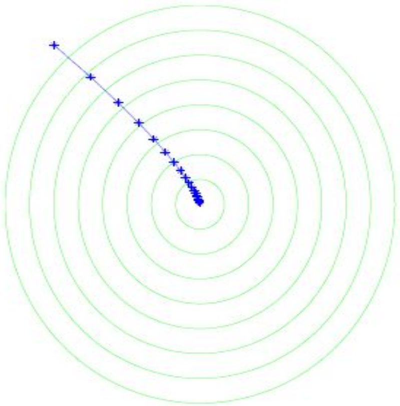
$$w_t = w_{t-1} - \eta \nabla f(w_{t-1})$$

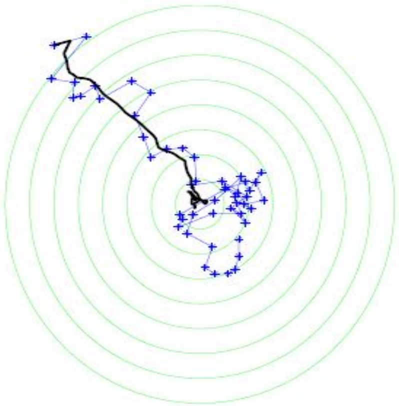
For a loss function, this is computed over a batch of n training samples:

$$\nabla l_w(x, y) = \frac{1}{n} \sum_{i=1}^{n} \nabla_w l_{w,t-1}(x_i, y_i)$$

In **stochastic gradient descent**, the gradient is estimated using a randomly-selected subset of the observations, or "minibatch" of m samples:

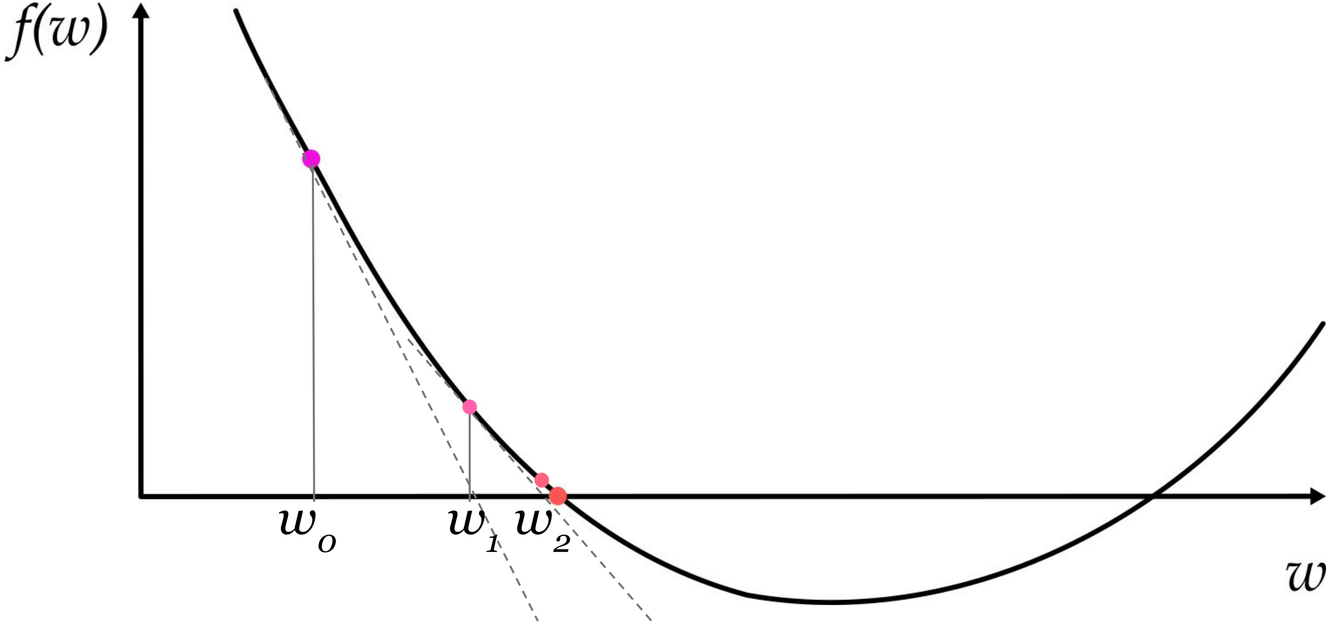
$$\nabla l_w(x, y) = \frac{1}{m} \sum_{i=1}^{n} \nabla_w l_{w,t-1}(x_i, y_i)$$





Understanding Machine Learning, Section 14.1, 14.3

Another approach: Newton's method



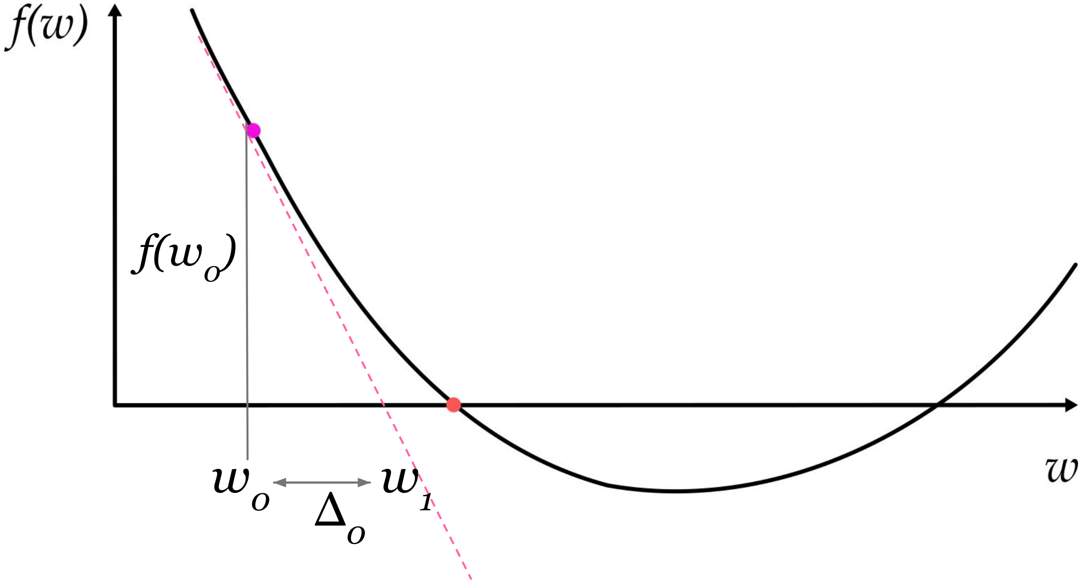
Deriving the update for Newton's method

$$w_1 = w_0 + \Delta_0$$

$$\begin{aligned}
f'(w_0) &= \frac{0 - f(w_0)}{w_1 - w_0} \\
&= \frac{-f(w_0)}{(w_0 + \Delta_0) - w_0} \\
&= -\frac{f(w_0)}{\Delta_0}
\end{aligned}$$

$$\implies \Delta_0 = -\frac{f(w_0)}{f'(w_0)}$$

$$\implies w_1 = w_0 - \frac{f(w_0)}{f'(w_0)}$$



Application of Newton's method to loss function minimization

Newton's method finds the roots of $f(w)$ via successive updates:

$$w_1 = w_0 - \frac{f(w_0)}{f'(w_0)}$$

In parameter estimation, we are interested the roots of $f(w) = \frac{dl_w}{dw} = l'_w(w)$. Therefore our update requires second-order information:

$$w_1 = w_0 - \frac{l'_w(w)}{l''_w(w)}$$

For a differentiable loss of more than one parameter $l_w : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$, this generalizes to

$$w_1 = w_0 - (\nabla^2 l_w)^{-1} \nabla l_w,$$

where $(\nabla^2 l_w)^{-1}$ is inverse of the Hessian.

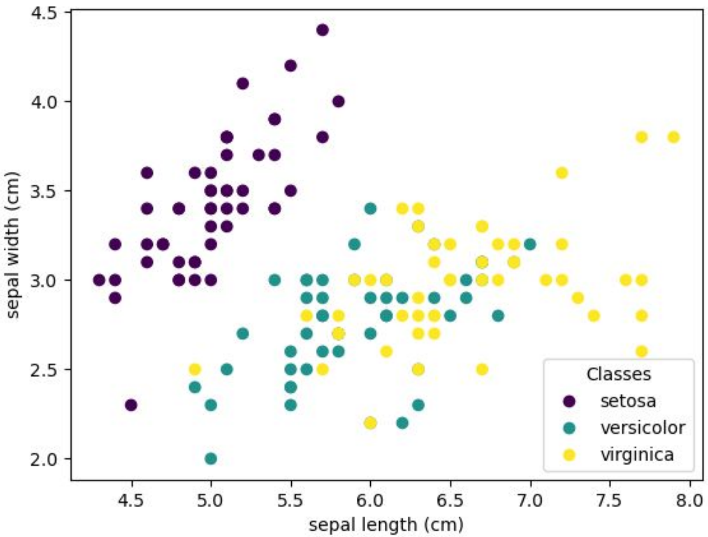
Key Questions

I. What is logistic regression?

II. How do we estimate the parameters?

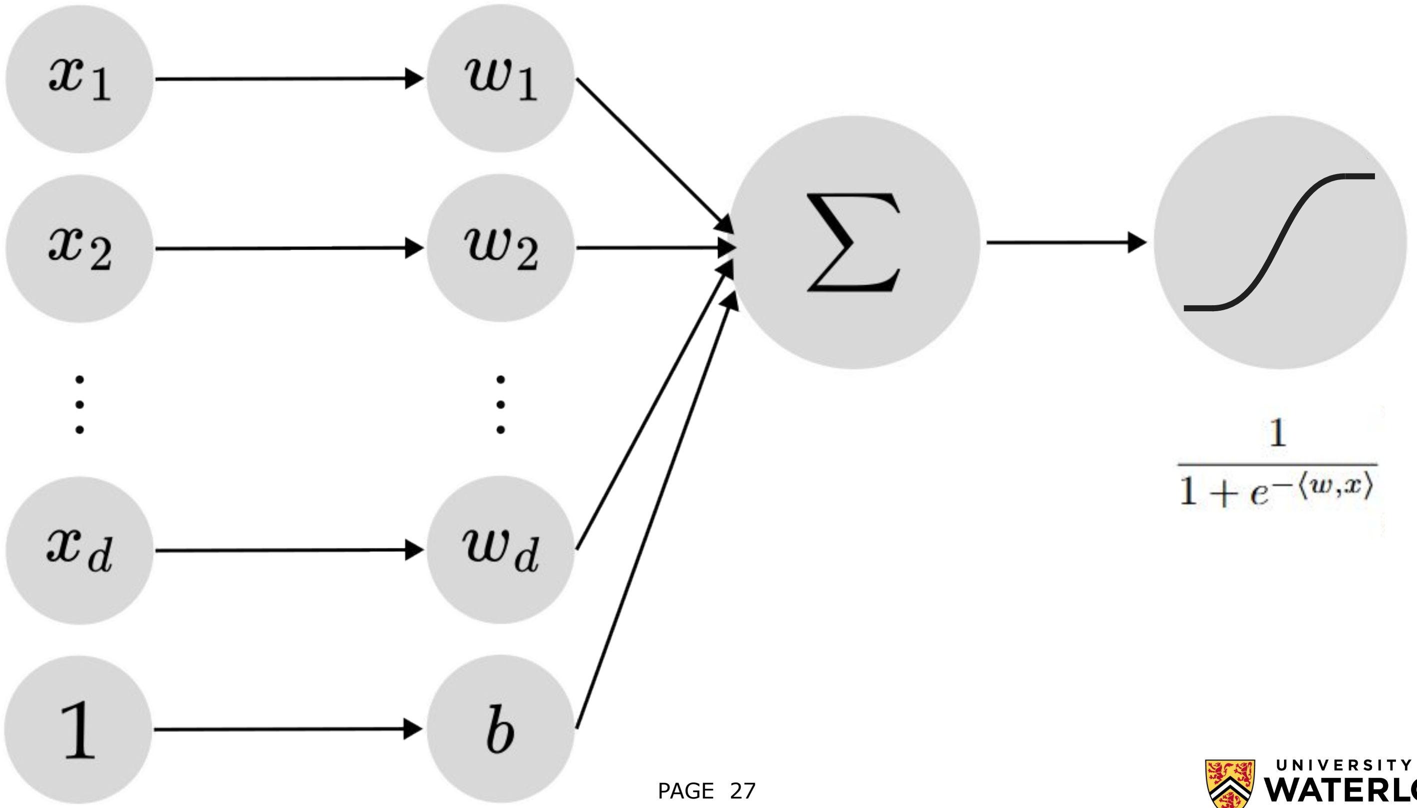
III. **How can we handle the multiclass case?**

Generalizing to the multiclass setting

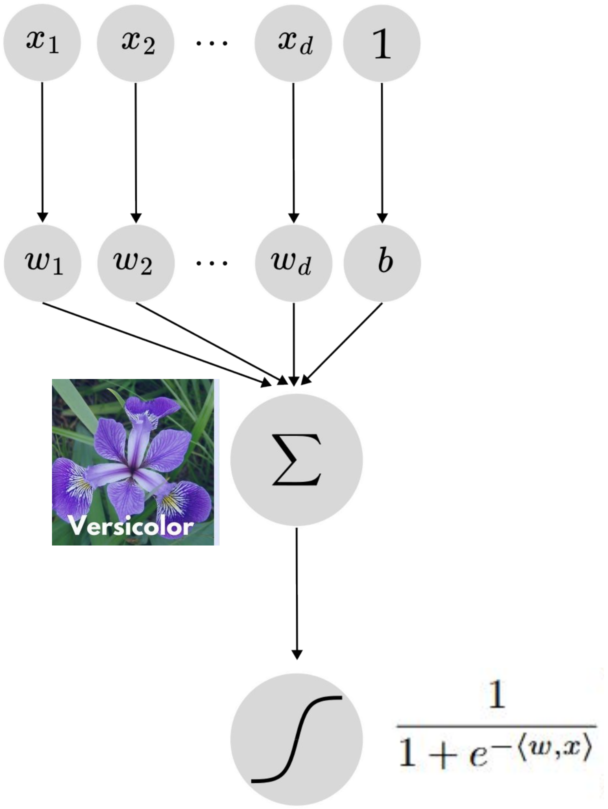


Figures: Sci-Kit Learn

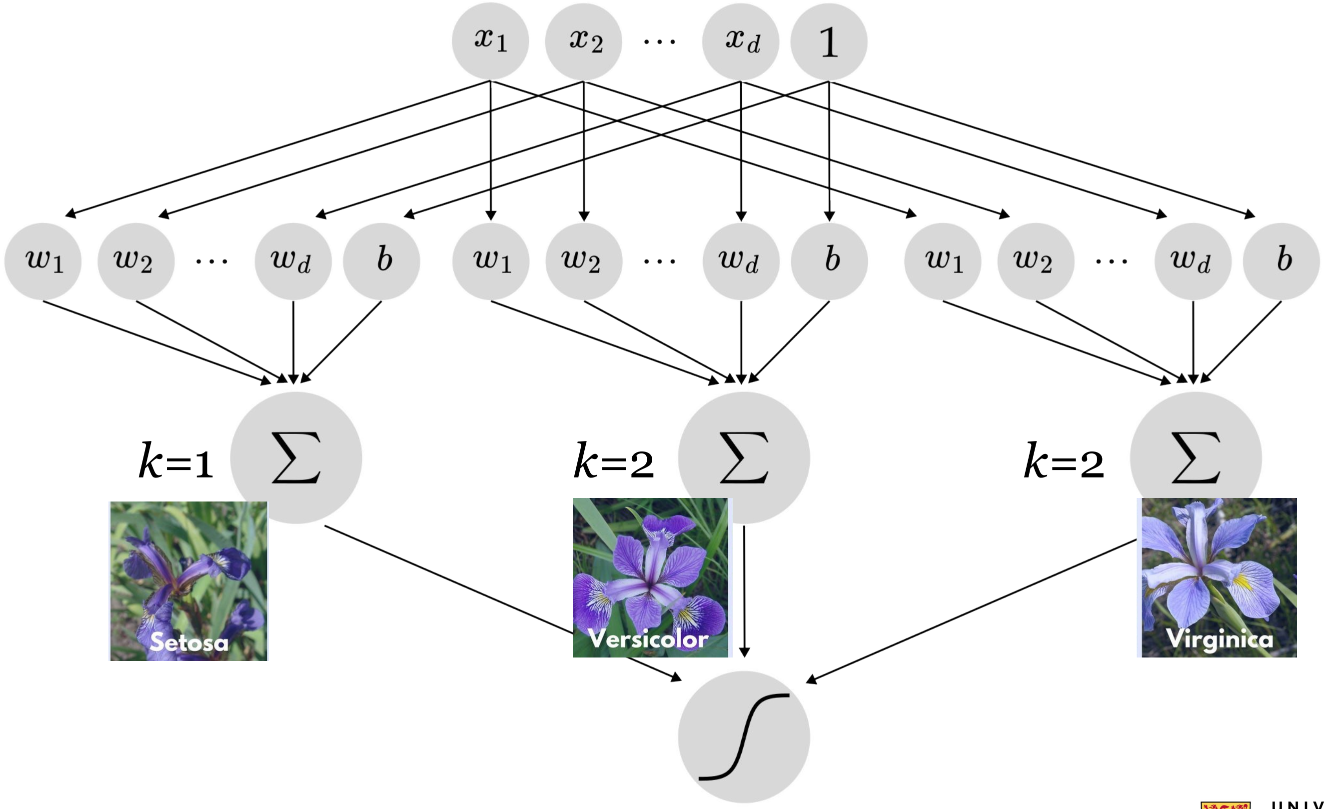
Architectural interpretation of logistic regression



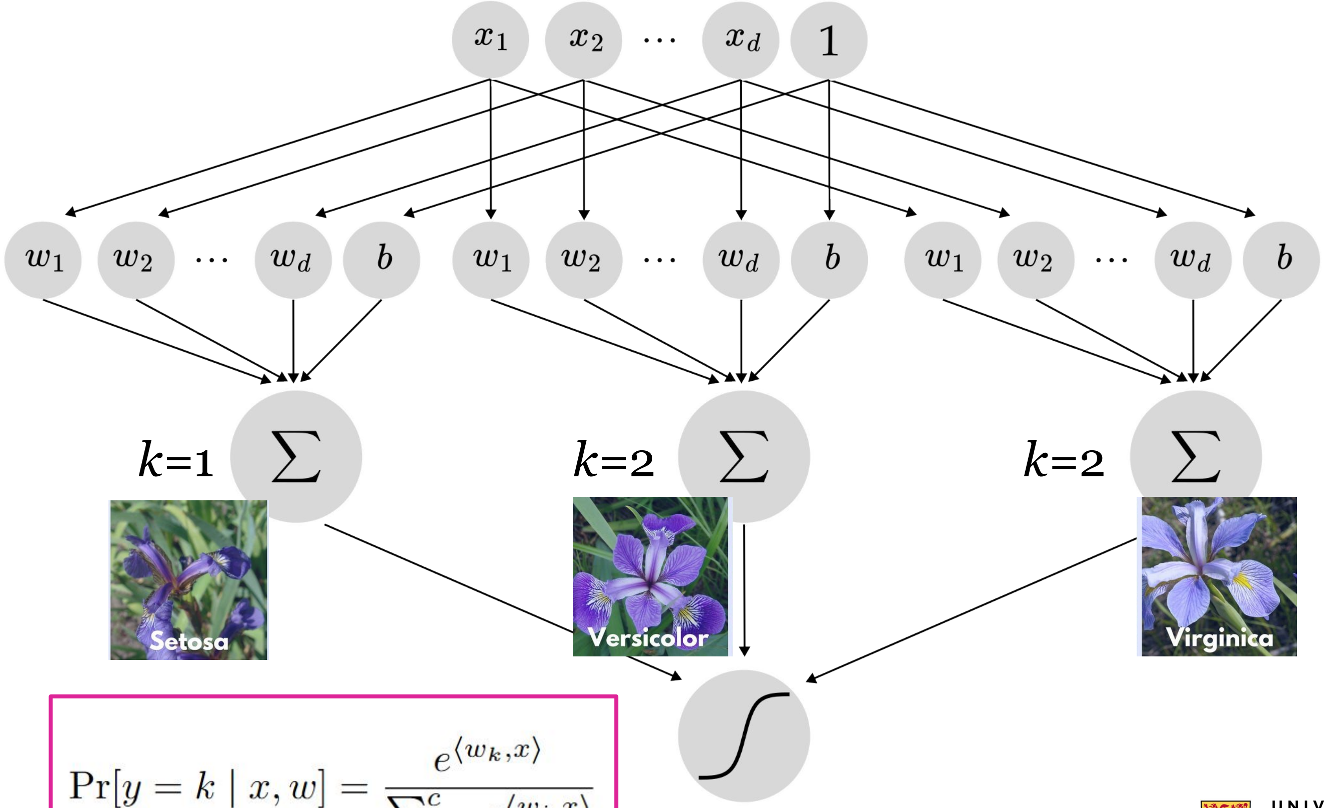
Logistic regression



Multinomial regression

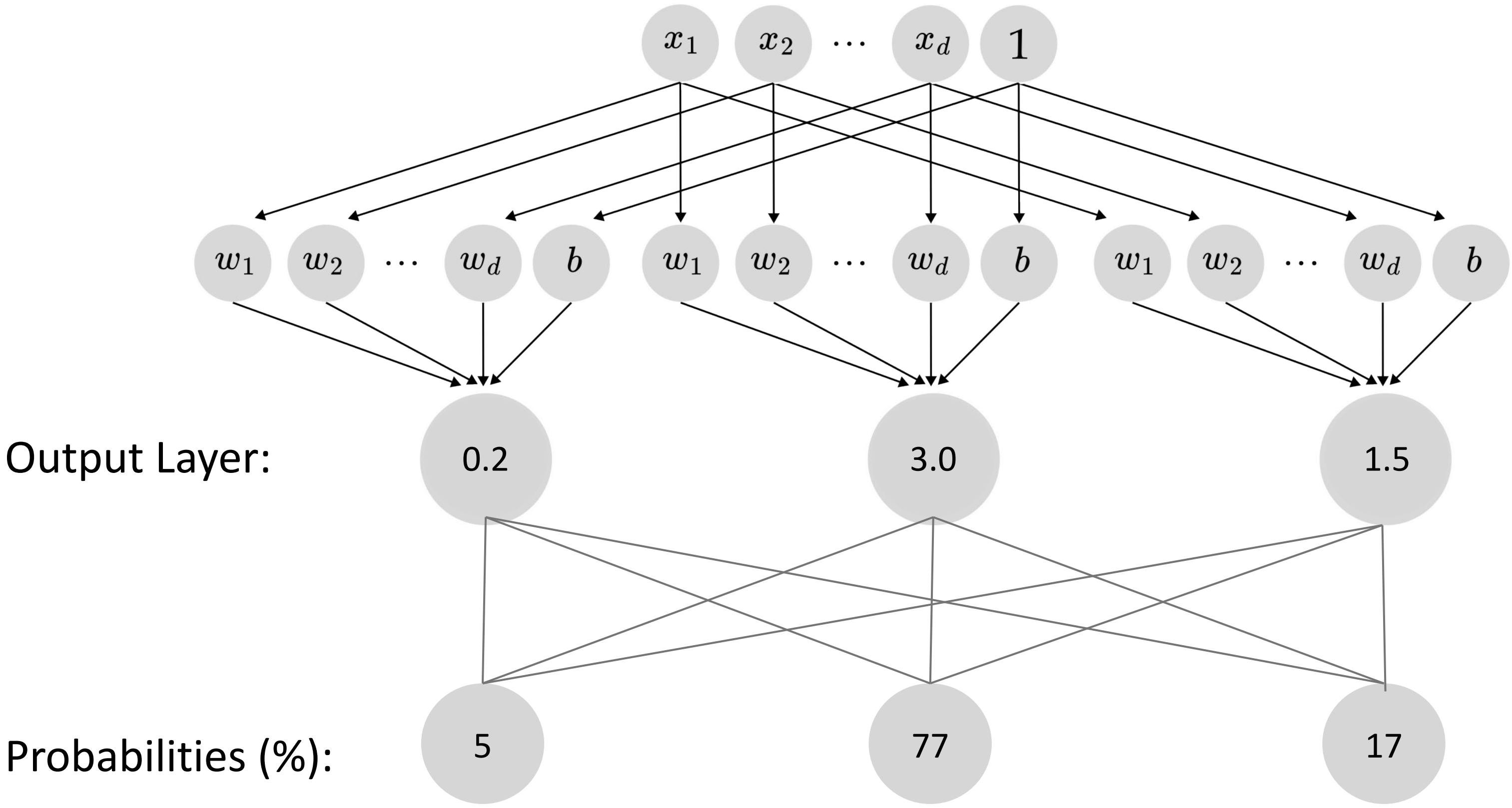


Multinomial regression



$$\Pr[y = k \mid x, w] = \frac{e^{\langle w_k, x\rangle}}{\sum_{i=1}^{c} e^{\langle w_i, x\rangle}}$$

Multinomial regression



Now that we're at the end of the lecture, you should be able to...

- ★ Recommend and justify application of logistic regression in appropriate **real-world scenarios**, as an alternative to linear regression and binary classification.
- ★ Explain the logistic regression **hypothesis class** using correct terminology, including conditional probability, sigmoid function, and linear predictor.
- ★ Sketch the **decision boundary** of a logistic regression predictor in a low-dimensional setting for different thresholds and parameters.
- ★ Defend the **cross-entropy loss** function used in logistic regression.
- ★ Explain the parametrization and hypothesis class of **multinomial regression** with reference to the **softmax function**.
- ★ Implement and apply **iterative optimization algorithms** including gradient descent, stochastic gradient descent, and the Newton-Raphson method.
- ★ Interpret the **meaning of coefficients** of a learned logistic regression model.

Similarly, if $y_i = 0$:

$$\log\mathcal{L}(w \mid \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \log\Big(1 - \frac{1}{1 + e^{-\langle w, x\rangle}}\Big)$$

$$= \sum_{i=1}^{n} \log\Big(\frac{e^{-\langle w, x\rangle}}{1 + e^{-\langle w, x\rangle}}\Big)$$

Using $\log\frac{a}{b} = \log a - \log b$

$$= \sum_{i=1}^{n} \log e^{-\langle w, x\rangle} - \log(1 + e^{-\langle w, x\rangle})$$

$$= \sum_{i=1}^{n} -\langle w, x\rangle - \log(1 + e^{-\langle w, x\rangle})$$

$$= \sum_{i=1}^{n} -\langle w, x\rangle - \log\Big(1 + \frac{1}{e^{\langle w, x\rangle}}\Big)$$

$$= \sum_{i=1}^{n} -\langle w, x\rangle - \log\Big(\frac{e^{\langle w, x\rangle} + 1}{e^{\langle w, x\rangle}}\Big)$$

Using again $\log\frac{a}{b} = \log a - \log b$

$$= \sum_{i=1}^{n} -\langle w, x\rangle - [\log(e^{\langle w, x\rangle} + 1) - \log e^{\langle w, x\rangle}]$$

$$= \sum_{i=1}^{n} -\langle w, x\rangle - \log(e^{\langle w, x\rangle} + 1) + \log e^{\langle w, x\rangle}$$

$$= \sum_{i=1}^{n} -\log(e^{\langle w, x\rangle} + 1)$$