Section 6. Logistic Regression

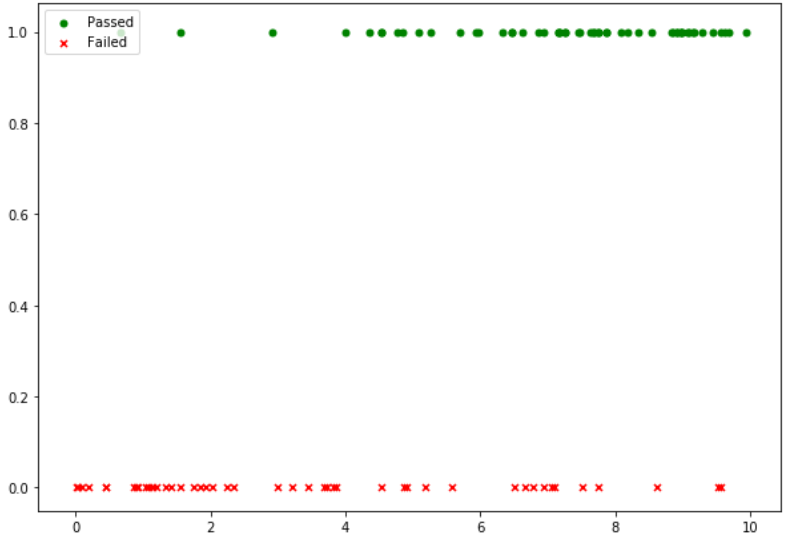
1. Logistic Regression (binary)
2. Softmax Regression (multiclass)

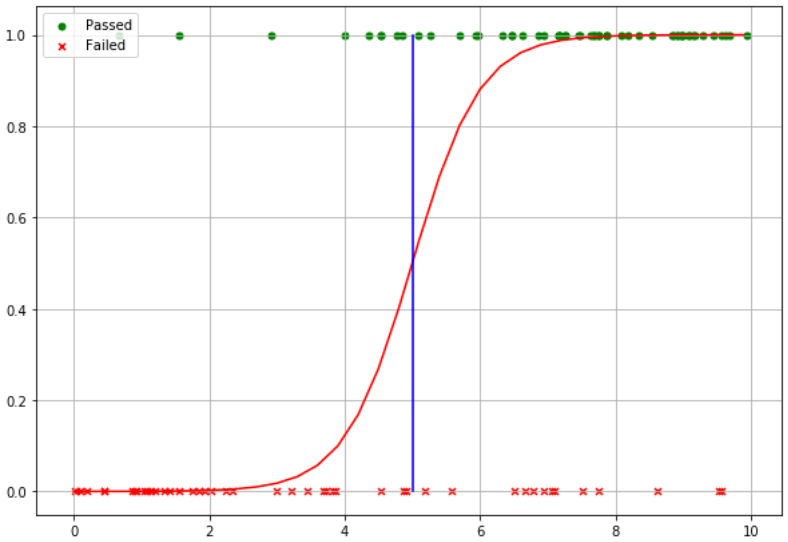
- Example: Data of students sleep time, study time, and pass/fail.

If we know the test scores, we can use linear regression to predict the test scores.

passed Y=1, failed Y=0

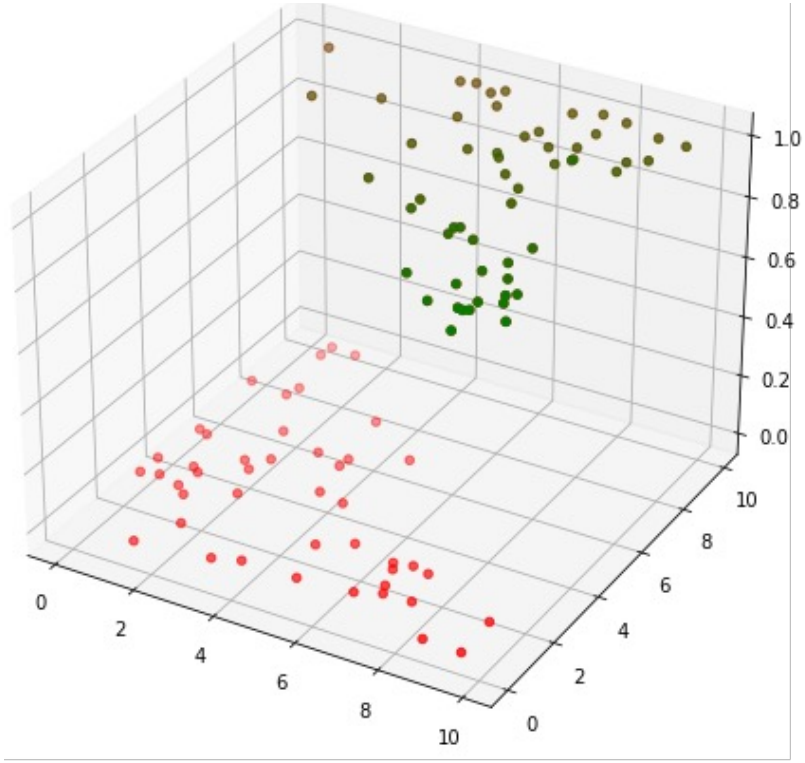
studied	Y
7.40	1
3.93	0
0.72	0
3.89	1
8.19	1
...	...

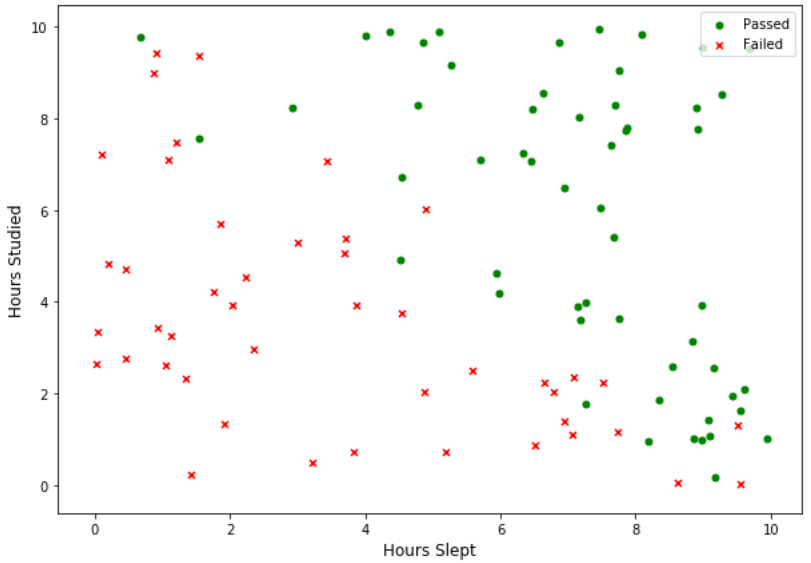




passed Y=1, failed Y=0

slept	studied	Y
7.63	7.40	1
2.03	3.93	0
3.82	0.72	0
7.15	3.89	1
6.47	8.19	1
...	...	...





➢ Logistic regression

Logistic regression is a **classification** algorithm, used to predict probabilities based on given set of independent variables.

Data: $D = \left\{\left(\vec{x}^{(i)}, y^{(i)}\right), i = 1, \ldots n\right\}$ $y^{(i)} \in \{0, 1\},$

Goal: Find conditional (posterior) probability

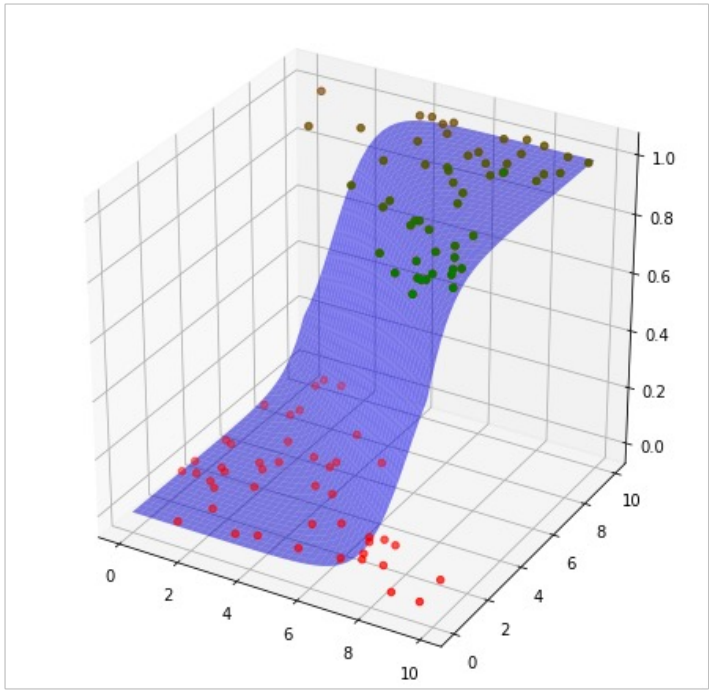
$$P(Y = k \,|\vec{X} = \vec{x}) \quad \text{for } k = 0, 1$$

➢ Bayes Decision Boundary

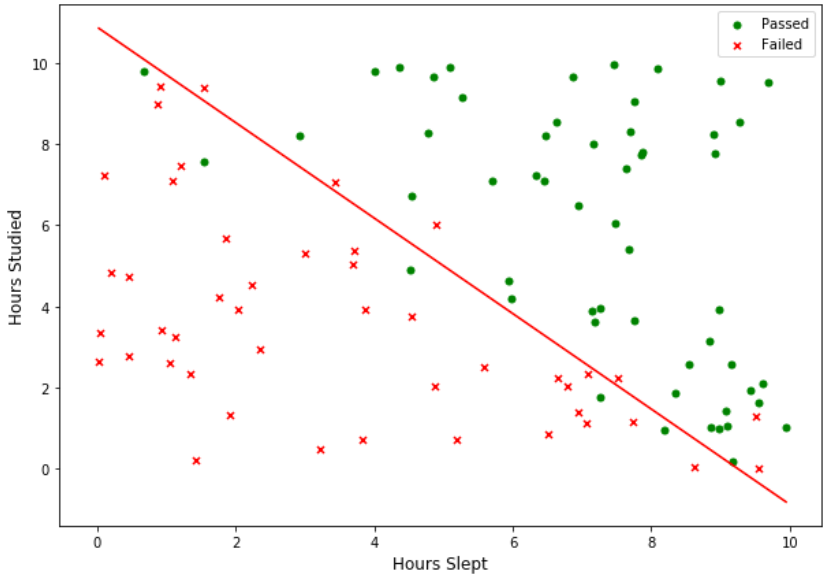
logistic regression prediction function returns a probability between 0 and 1, in order to predict which class this data belongs we need to set a threshold.

Bayes Boundary $P(Y = 0 \mid \vec{x}) = P(Y = 1 \mid \vec{x})$

Or $P(Y = 1 \mid \vec{x}) = 0.5$



$$h_{\vec{\theta}}(\vec{x}) = P(Y = 1 \mid \vec{x})$$

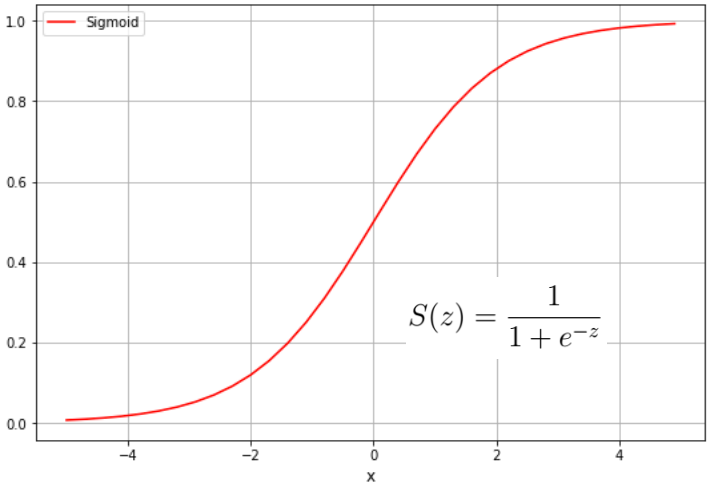


$$h_{\vec{\theta}}(\vec{x}) = 0.5$$

Logistics regression.

The **sigmoid function** maps any real value into a value in [0,1].

$$S(z) = \frac{1}{1 + e^{-z}}$$



- **Logistics regression assumption:**

$$P(Y = 1 \mid \vec{x}) \coloneqq h_{\vec{\theta}}(\vec{x}) \coloneqq S\left(\vec{\theta}^T \vec{x}\right) = \frac{1}{1 + e^{-\vec{\theta}^T \vec{x}}}$$

- **Prediction**:

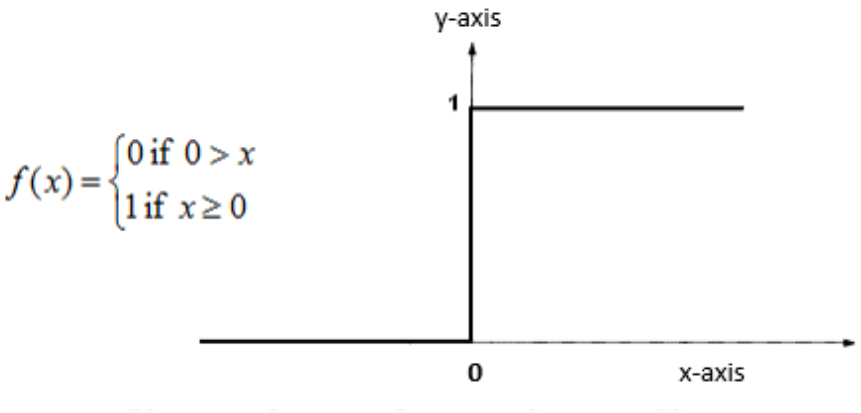
$$C(\vec{x}) = \begin{cases} 1, & if\ h(\vec{x}) \geq 0.5 \\ 0, & if\ h(\vec{x}) < 0.5 \end{cases}$$

- **Bayes Decision Boundary**

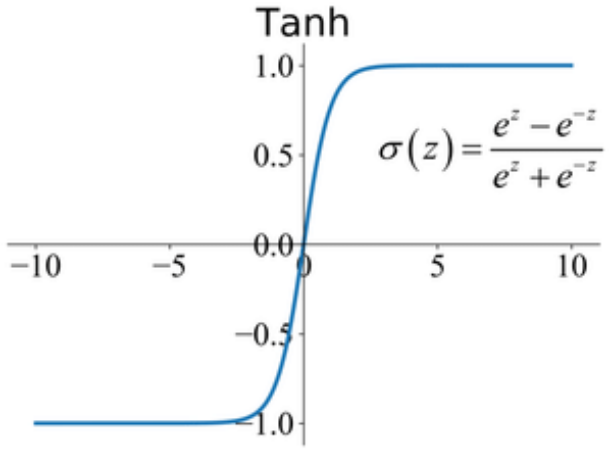
$$\vec{\theta}^T \vec{x} = 0$$

- Other activation functions

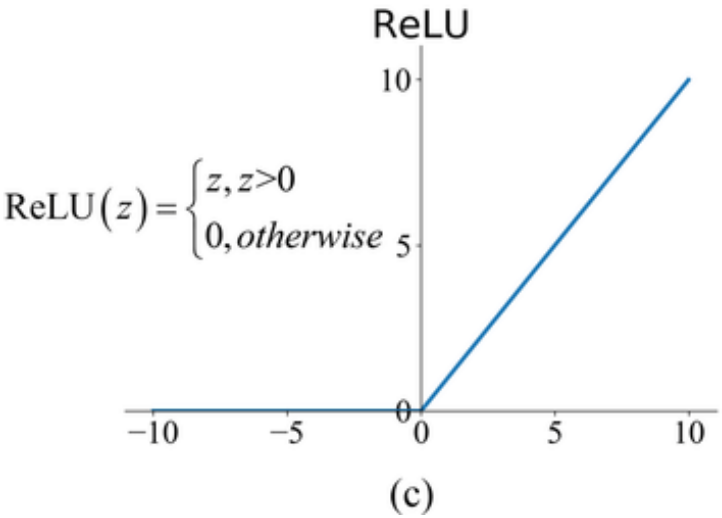
Step



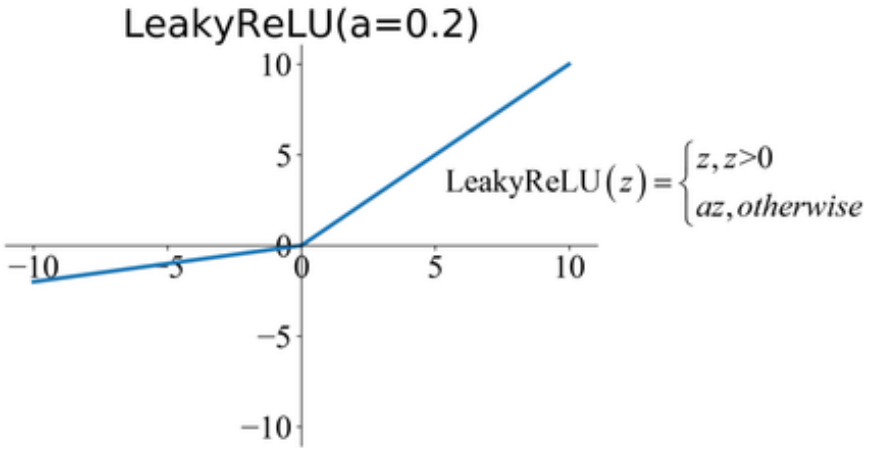
(a)



(b)



(c)



(d)

Rectified Linear Unit (ReLU)

➢ Maximize Likelihood method:

Logistics regression Assumption (with label space $\mathcal{C} = \{0, 1\}$):

$$P(Y = 1 \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x})$$

$$P(Y = 0 \mid \vec{x}; \vec{\theta}) = 1 - h_{\vec{\theta}}(\vec{x})$$

Equivalently,

$$P(Y = y \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x})^{y} \left(1 - h_{\vec{\theta}}(\vec{x})\right)^{1-y}$$

The above random variable Y is the **Bernoulli Distribution** with probability $p = h_{\vec{\theta}}$ depending on $\vec{x}$ and parameter $\vec{\theta}$.

Given labeled data: $(\boldsymbol{X}, \vec{\boldsymbol{y}})$ $\quad y^{(i)} \in \{0, 1\}$

Likelihood function:

$$L(\vec{\theta}) = P(\vec{y} \mid X; \vec{\theta})$$

$$= \prod_{i=1}^{n} P\left(y^{(i)} \mid \vec{x}^{(i)} ; \vec{\theta} \right)$$

$$= \prod_{i=1}^{n} h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)^{y^{(i)}} \left(1 - h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)\right)^{1-y^{(i)}}$$

Log Likelihood function:

$$l(\vec{\theta}) = \log L(\vec{\theta})$$

$$= \sum_{i=1}^{n} \left(y^{(i)} \ln h_{\vec{\theta}}\left(\vec{x}^{(i)}\right) + (1 - y^{(i)}) \ln \left(1 - h_{\vec{\theta}}\left(\vec{x}^{(i)}\right)\right)\right)$$

Optimization: (Maximize Likelihood)

$$\text{argmax } L(\vec{\theta})$$

$$= \text{argmax } l(\vec{\theta})$$

$$= \text{arg}\mathbf{min} \; -\frac{\mathbf{1}}{\mathbf{n}} \sum_{i=1}^{n} \left(y^{(i)} \ln h_{\vec{\theta}}(\vec{x}^{(i)}) + (1 - y^{(i)}) \ln \left(1 - h_{\vec{\theta}}(\vec{x}^{(i)})\right)\right)$$

Cross-entropy Loss $J(\vec{\theta})$

Or log-cost function

Cost for each individual point $\vec{x}^{(i)}, y^{(i)}$:

$$J(\vec{\theta}; \vec{x}^{(i)}) = \begin{cases} -\ln h_{\vec{\theta}}(\vec{x}^{(i)}) & if \;\; y^{(i)} = 1 \\ -\ln \left(1 - h_{\vec{\theta}}(\vec{x}^{(i)})\right) & if \;\; y^{(i)} = 0 \end{cases}$$

- **Gradient descent for Cross-entropy Loss**

$$\nabla J(\vec{\theta}) = \begin{bmatrix} \frac{\partial J(\vec{\theta})}{\partial \theta_0} \\ \vdots \\ \frac{\partial J(\vec{\theta})}{\partial \theta_d} \end{bmatrix} \qquad \frac{\partial J(\vec{\theta})}{\partial \theta_j} = ?$$

Recall: $h_{\vec{\theta}}(\vec{x}) := S(\vec{\theta}^T \vec{x}) = \frac{1}{1 + e^{-\vec{\theta}^T \vec{x}}}$

$$\frac{d\, S(z)}{d\, z} = S(z)\big(1 - S(z)\big)$$

$$\frac{\partial h_{\vec{\theta}}(\vec{x}^{(i)})}{\partial \theta_j} = S(z)\big(1 - S(z)\big)x_j^{(i)} \qquad z = \vec{\theta}^T \vec{x}$$

$$\frac{\partial J(\vec{\theta})}{\partial \theta_j} = -\frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)}\frac{1}{S(z)}\,S(z)\big(1-S(z)\big)x_j^{(i)} - \big(1-y^{(i)}\big)\frac{1}{1-S(z)}\,S(z)\big(1-S(z)\big)x_j^{(i)}\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\big(S\big(\vec{\theta}^T\vec{x}\big) - y^{(i)}\big)x_j^{(i)}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\big(h\big(\vec{x}^{(i)}\big) - y^{(i)}\big)x_j^{(i)}$$

Vector notation of the gradient:

$$\nabla_{\vec{\theta}}\, J = \frac{1}{n}X^T\big(h_{\vec{\theta}}(X) - \vec{y}\,\big)$$

➢ Gradient Descent and Newton's method for Logistics Regression

- **Gradient Descent:**

$$\vec{\theta}_{k+1} = \vec{\theta}_k - \alpha \nabla_{\vec{\theta}_k} J = \vec{\theta}_k - \alpha \frac{1}{n} X^T \left(h_{\vec{\theta}_k}(X) - \vec{y} \right)$$

- **Newton's method:**

$$\vec{\theta}_{k+1} = \vec{\theta}_k - H^{-1} \nabla J(\vec{\theta}_k)$$

Here H is the Hessian matrix $H = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1{}^2} & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_d \partial \theta_1} & \cdots & \frac{\partial^2 J}{\partial \theta_d{}^2} \end{bmatrix}$

with $H_{jk} = \frac{\partial^2 J}{\partial \theta_j \partial \theta_k} = \frac{1}{n} \sum_{i=1}^{n} h\left(\vec{x}^{(i)}\right) \left(1 - h\left(\vec{x}^{(i)}\right)\right) x_j^{(i)} x_k^{(i)}$

Matrix Notation for $H = \frac{1}{n} X^T A X$, where A=**diag** $\left[h\left(\vec{x}^{(i)}\right) \left(1 - h\left(\vec{x}^{(i)}\right)\right) \right]$

Question: If $y \in \{-1, 1\}$,

$$P(Y = 1 \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(\vec{x})$$

$$P(Y = -1 \mid \vec{x}; \vec{\theta}) = 1 - h_{\vec{\theta}}(\vec{x})$$

Equivalently,

$$P(Y = y \mid \vec{x}; \vec{\theta}) = h_{\vec{\theta}}(y\vec{x}) \qquad \text{(why?)}$$

1. Find $J(\vec{\theta})$.
2. Calculate gradient $\nabla_{\vec{\theta}} J$
3. Calculate Hessian matrix.

Odds Ratio: A ratio of two probabilities.

Log Odd Ratio: logarithm of an odds ratio.

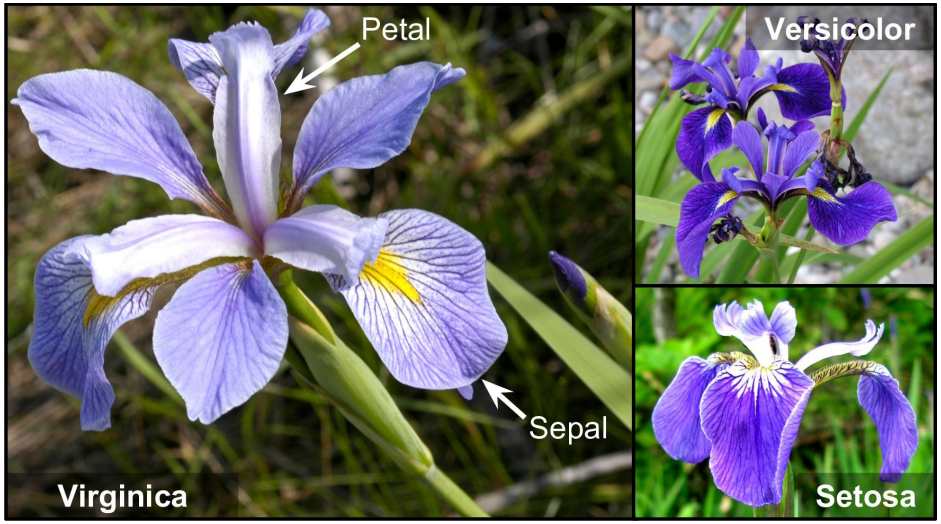
$$\log \frac{P(Y=1|\vec{x})}{P(Y=0|\vec{x})} = \log \frac{h(\vec{x})}{1-h(\vec{x})} := \vec{\theta}^T \vec{x}$$

Logistic Regression assumption $h_{\vec{\theta}}(\vec{x}) := \frac{1}{1+e^{-\vec{\theta}^T \vec{x}}}$

Softmax Regression (Multinomial Logistic Regression)

- Flowers of three iris plant species:

The famous Iris database, first used by Sir R.A. Fisher(1936), is best known database to be found in the pattern recognition literature. It contains the **sepal** and **petal length** and **width** of 150 iris flowers of three different species: Iris-Setosa, Iris-Versicolor, and Iris-Virginica.



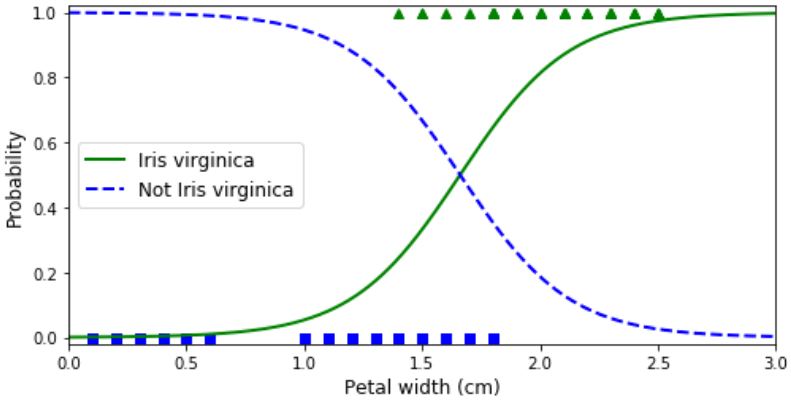
Data features:
Sepal length
Sepal width
Petal length
Petal width

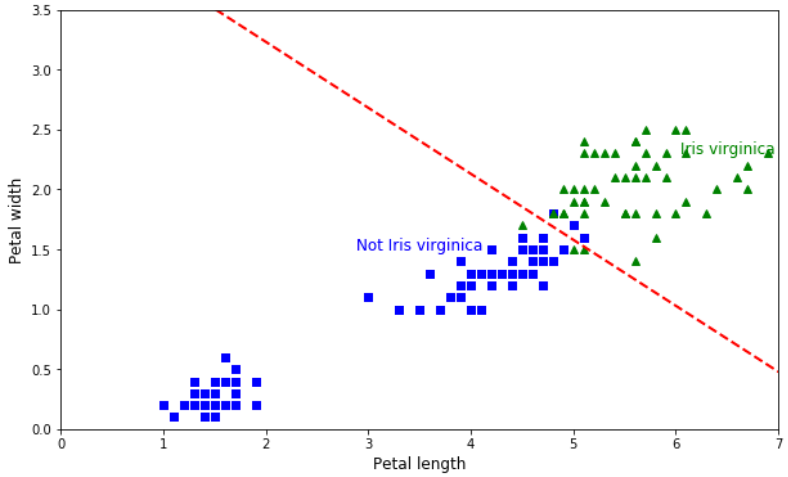
Classes: 0-Iris-Setosa, 1-Iris-Versicolour, 2-Iris-Virginica

Data: $D = \{(\vec{x}^{(i)}, y^{(i)}), i = 1, \dots n\}$ $\quad y^{(i)} \in \{0, 1, 2\}$

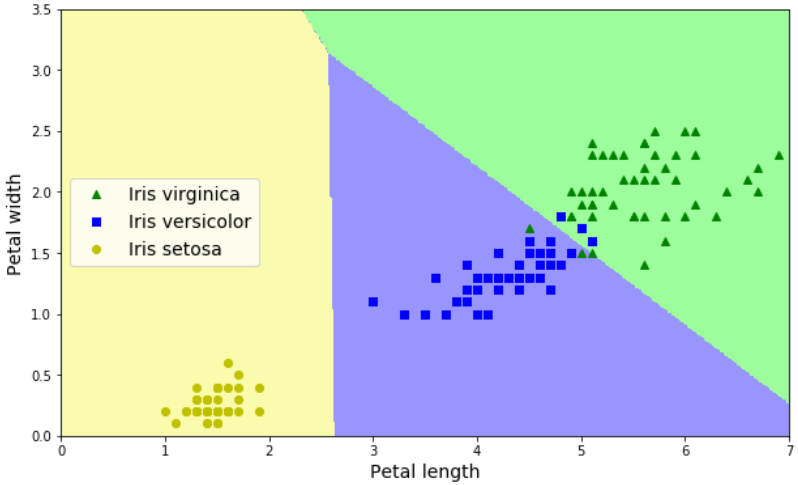
[5.1, 3.5, 1.4, 0.2]
[4.9, 3. , 1.4, 0.2]
[4.7, 3.2, 1.3, 0.2]
[4.6, 3.1, 1.5, 0.2]
[5. , 3.6, 1.4, 0.2]
[5.4, 3.9, 1.7, 0.4]
[4.6, 3.4, 1.4, 0.3]
[5. , 3.4, 1.5, 0.2]
[4.4, 2.9, 1.4, 0.2]
...

one v.s. rest





Softmax:



Softmax Regression

Goal:

$$P(Y = k \mid \vec{X} = \vec{x}) = ? \qquad \text{for } k = 0, 1, \dots, K$$

Assumption:

$$\begin{bmatrix} P(Y = 0 \mid \vec{x};\ \vec{\theta}) \\ P(Y = 1 \mid \vec{x};\ \vec{\theta}) \\ P(Y = 2 \mid \vec{x};\ \vec{\theta}) \end{bmatrix} := \frac{1}{\sum_{j=0}^{K} \exp \vec{\theta}_j{}^T \vec{x}} \begin{bmatrix} \exp \vec{\theta}_0{}^T \vec{x} \\ \exp \vec{\theta}_1{}^T \vec{x} \\ \exp \vec{\theta}_2{}^T \vec{x} \end{bmatrix} =: h_{\vec{\theta}}(\vec{x})$$

Here $\vec{\theta}_j = \begin{bmatrix} \theta_{j,0} \\ \theta_{j,1} \\ \vdots \\ \theta_{j,d} \end{bmatrix}$

So, we have $K(d+1)$ parameters $\Theta = [\vec{\theta}_1 \ \dots \vec{\theta}_K]$.

Cross-entropy (log-cost) Loss

$$J(\vec{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} \mathbb{I}\left(y^{(i)} = j\right) \ln P\left(y^{(i)} = j \middle| \vec{x}^{(i)}; \vec{\theta}\right)$$

$$= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} \mathbb{I}\left(y^{(i)} = j\right) \ln \frac{\exp \vec{\theta}_j{}^T \vec{x}^{(i)}}{\sum_{l=0}^{K} \exp \vec{\theta}_l{}^T \vec{x}^{(i)}}$$

$\mathbb{I}(\ )$ is the **indicator function**:

$$\mathbb{I}(\text{True}) = 1$$

$$\mathbb{I}(\text{False}) = 0$$

- **Gradient Descent:**

The **gradient** of Cross-entropy Loss is

$$\nabla_{\vec{\theta}_j} J(\vec{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left(h_{\vec{\theta}}(\vec{x}^{(i)}) - \mathbb{I}(y^{(i)} = j) \right) \vec{x}^{(i)}$$

Gradient Descent:

$$\vec{\theta}^{next} = \vec{\theta} - \alpha \nabla_{\vec{\theta}} J$$

Hessian is non-invertible in this case, so we can not use Newton's method directly.

➢ Some Remarks:

- Logistics regression with **non-linear** boundaries:

Similarly, as linear regression, we can introduce new features

$$z_1 = x_1, z_2 = x_2, z_3 = x_1^2, z_4 = x_2^2, z_5 = x_1 x_2, z_6 = x_1^3,$$

$$z_7 = x_2^3, z_8 = x_1^2 x_2, z_9 = x_1 x_2^2, \ldots$$

Apply logistics regression to the new features, get the boundary and replace back to $x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^3, x_2^3, x_1^2 x_2, x_1 x_2^2 \ldots$

Then we get the non-linear boundary.

- Logistic regression with (ridge/lasso) regularization

Regularization Cost = **Cross-entropy Loss +Penalty**

$$J^{ridge}(\vec{\theta}) = J(\vec{\theta}) + \lambda \sum_{j=1}^{d} {\theta_j}^2$$

$$J^{lasso}(\vec{\theta}) = J(\vec{\theta}) + \lambda \sum_{j=1}^{d} |\theta_j|$$

Convert Categorical Data to Numerical Data

We used **Integer Encoding** for the classification, which means using 0,1,…, K for classes.

Note that in a K-class classification the individual classes can sometimes be usefully represented as K-length binary variables. (**One-Hot Encoding**)

This means we denote class j to be

$$\vec{e}_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \in \mathbb{R}^K$$

The binary variables are often called “dummy variables” in statistics.

- **Applications**:

1. Email spam detector
2. Diagnose a person with a set of syndromes as virus carrier or non-carrier.
3. Identify which gene, out of a million genes, is disease-causing or not.
4. Judge if a trading activity is a fraud or not.
5. …