

# **Advanced Analytical Theory and Methods: Logistic Regression**

**Prof. Dr. Shamim Akhter**

**Professor, Dept. of CSE**

**Ahsanullah University of Science and Technology**

# Regression vs. Classification

- Regression:
  - Regression predictive modeling is the task of approximating a mapping function ( $f$ ) from input variables ( $X$ ) to a **continuous output variable** ( $y$ ).
    - Continuous output variable
      - real-value, such as an integer or floating point value.
      - quantities, such as amounts and sizes.
      - For example, a house may be predicted to sell for a specific dollar value, perhaps in the range of \$100,000 to \$200,000.
  - A problem with multiple input variables is often called a multivariate regression problem.
  - A regression problem where input variables are ordered by time is called a time series forecasting problem.

# Regression vs. Classification

- Classification:

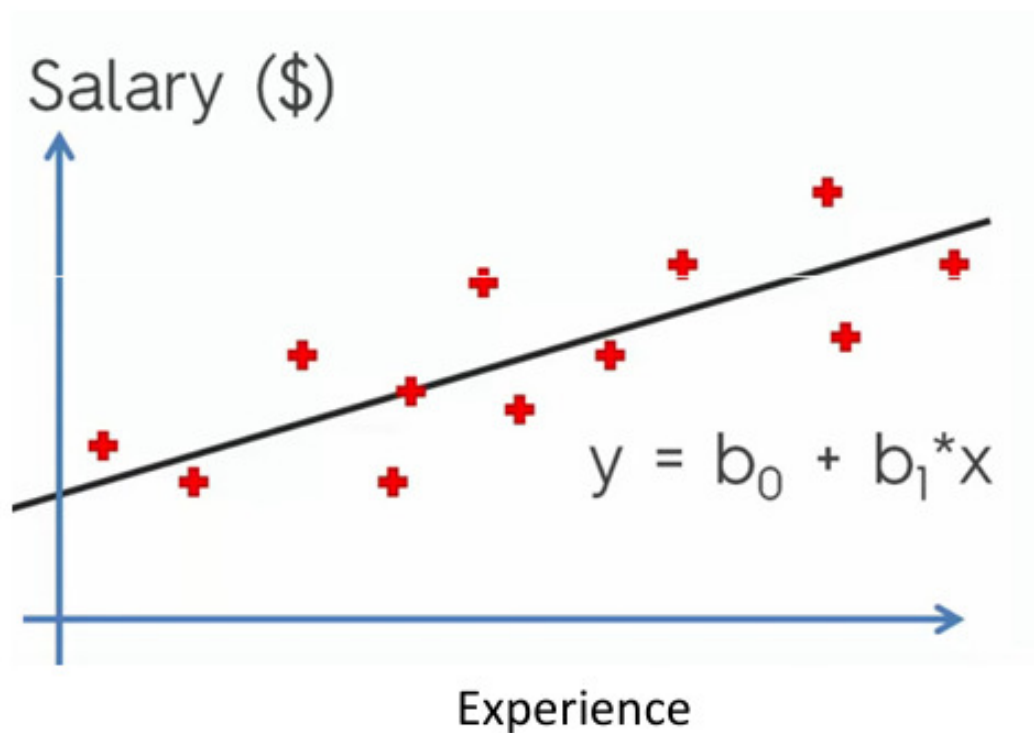
- Classification predictive modeling is the task of approximating a mapping function ( $f$ ) from input variables ( $X$ ) to **discrete output variables** ( $y$ ).
  - Discrete Output Variable
    - The mapping function predicts the class or category for a given observation, often called labels or categories.
    - For example, an email of text can be classified as belonging to one of two classes: “spam” and “*not spam*”.
- A problem with two classes is often called a two-class or binary classification problem.
- A problem with more than two classes is often called a multi-class classification problem.
- A problem where an example is assigned multiple classes is called a multi-label classification problem.

# Logistic Regression

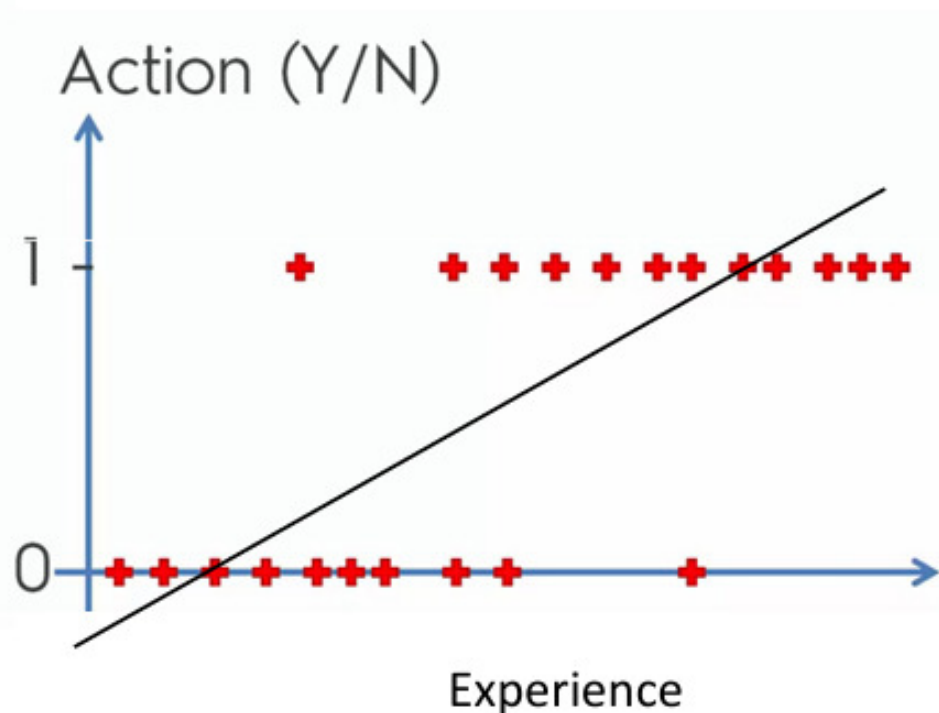
- Logistic regression:
  - a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome.
  - The outcome is measured with a **dichotomous variable**
    - in which there are only two possible outcomes.
- Logistic regression is mostly used as **1-layer classifier network** to the binary classification problems.

# Linear vs. Logistic

We Know Already



New problem



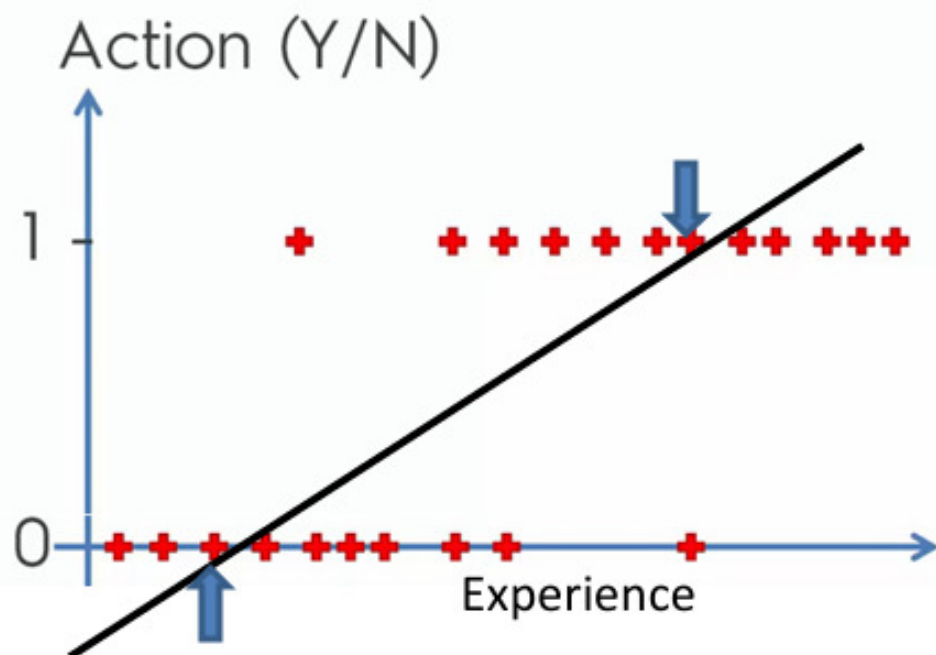
Low Residual  $R^2$

# Can't we use Linear Regression for Classification?

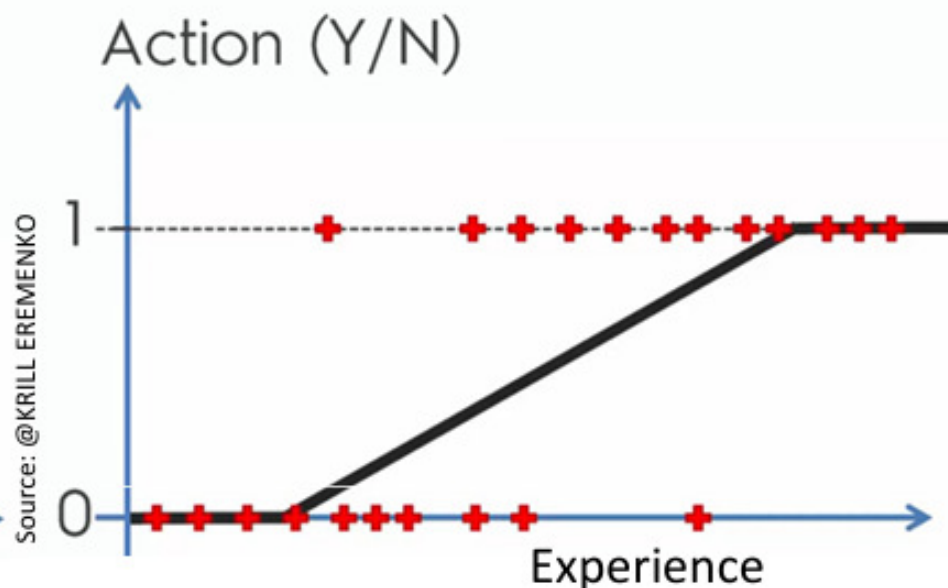
- Multiclass classification
- Binary classification[0,1]
  - Predicted values may be out of range
- Variance is not constant
- The normality assumption
  - Residual errors are normally distributed.

Although there may be settings where using linear regression to model a binary outcome may not lead to ruin, in general it is not a good idea. Essentially doing so (usually) amounts to using the wrong tool for the job.

# Logistic Regression



Action value is in between 0 and 1  
Probability is also between 0 and 1  
and consider action as chance



Source: @KRILL EREMENKO

## Lesson Learned:

1.  $F(X)$  must always be positive (Since  $P \geq 0$ )
2.  $F(X)$  must be less than 1 (Since  $P \leq 1$ )
3. Constrain outcomes  $P$  such that  $0 \leq P \leq 1$



# Logistic Regression

- $F(X)$  must always be positive (Since  $P \geq 0$ )

$$P = e^{(b_0 + b_1 X)}$$

Satisfy the lower bound ( $\geq 0$ )  
but do not satisfy upper bound ( $\leq 1$ )

- $F(X)$  must less than 1 (Since  $P \leq 1$ )

$$P = \frac{e^{(b_0 + b_1 X)}}{e^{(b_0 + b_1 X)} + 1}$$

Very large  $e^{(b_0 + b_1 X)}$   $P \sim 1$   
Very small  $e^{(b_0 + b_1 X)}$   $P \sim 0$

Log Sigmoid Function

**But this Logistic Function is Nonlinear!**

**Logistic Regression requires Linear Function.**



# Logistic Regression

- Probability of success  $P(Y=1)=P = \frac{e^{(b_0+b_1X)}}{e^{(b_0+b_1X)} + 1}$
- Probability of failure  $P(Y=0)=1-P=1 - \frac{e^{(b_0+b_1X)}}{e^{(b_0+b_1X)} + 1}$

$$\text{Odds Ratio} = \frac{P}{1-P} = \frac{\frac{e^{(b_0+b_1X)}}{e^{(b_0+b_1X)} + 1}}{1 - \left( \frac{e^{(b_0+b_1X)}}{e^{(b_0+b_1X)} + 1} \right)}$$

An odds ratio (OR) is a measure of association between an exposure and an outcome.

- Odds Ratio (relative risk) measures the probability that  $Y=1$  relative to the probability that  $Y=0$ 
  - Example : Odds Ratio of 2 >> means
  - the outcome  $Y=1$  is 2x as likely as outcome  $Y=0$

# Logistic Regression

- Derive from the Odds Ratio model we find

$$\frac{P}{1-P} = e^{(b_0 + b_1 X)}$$

Apply natural log on both side we get the following logistic regression model

$$\ln\left(\frac{P}{1-P}\right) = b_0 + b_1 X$$

Thus, even though  $P$  is not a linear function of  $X$ , but this transformation is a linear function of  $X$

# Logistic Regression

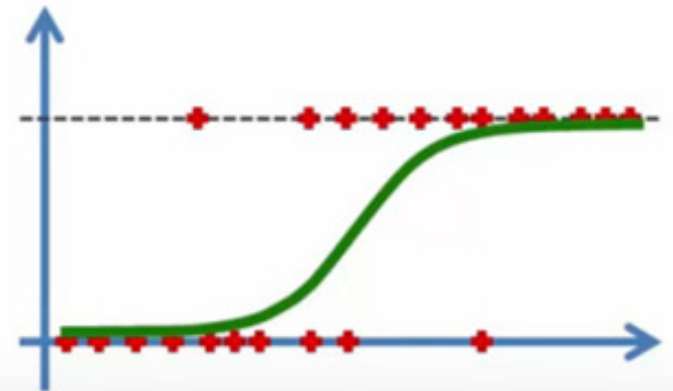
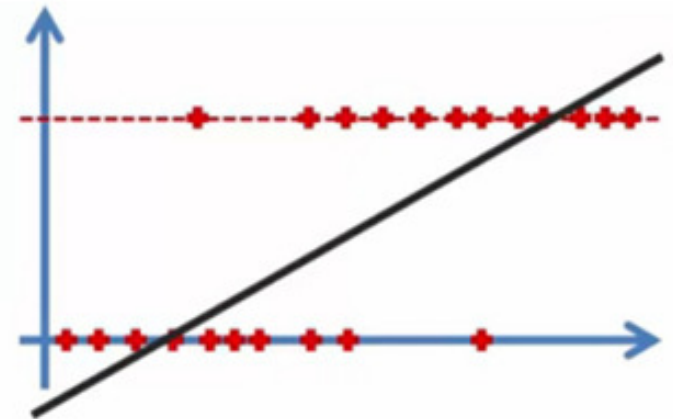
$$y = b_0 + b_1 * x$$

Sigmoid Function

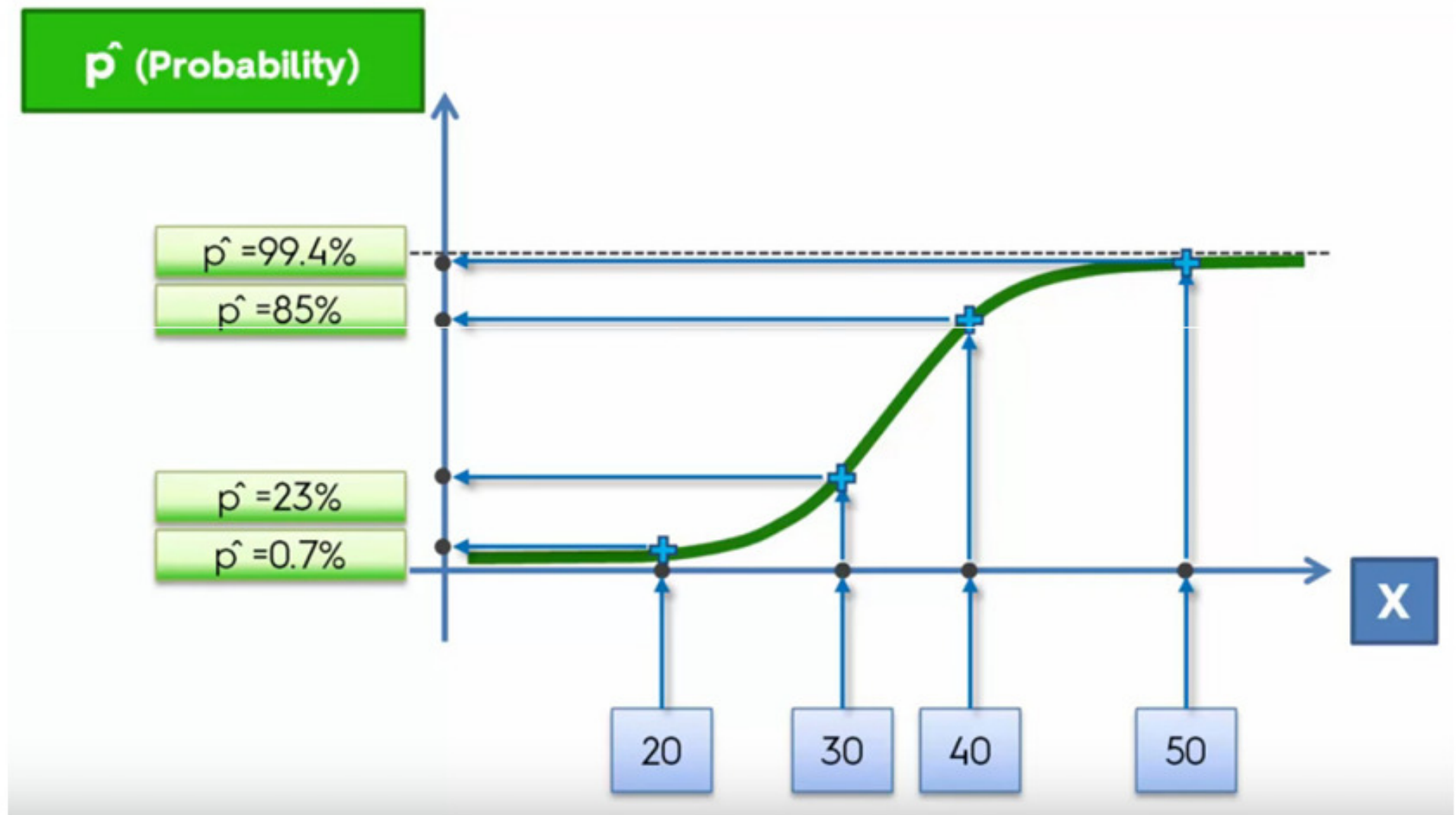
$$p = \frac{1}{1 + e^{-y}}$$

Where e is the base of the natural logarithms (Euler's number)

$$\ln \left( \frac{p}{1 - p} \right) = b_0 + b_1 * x$$



# Logistic Regression



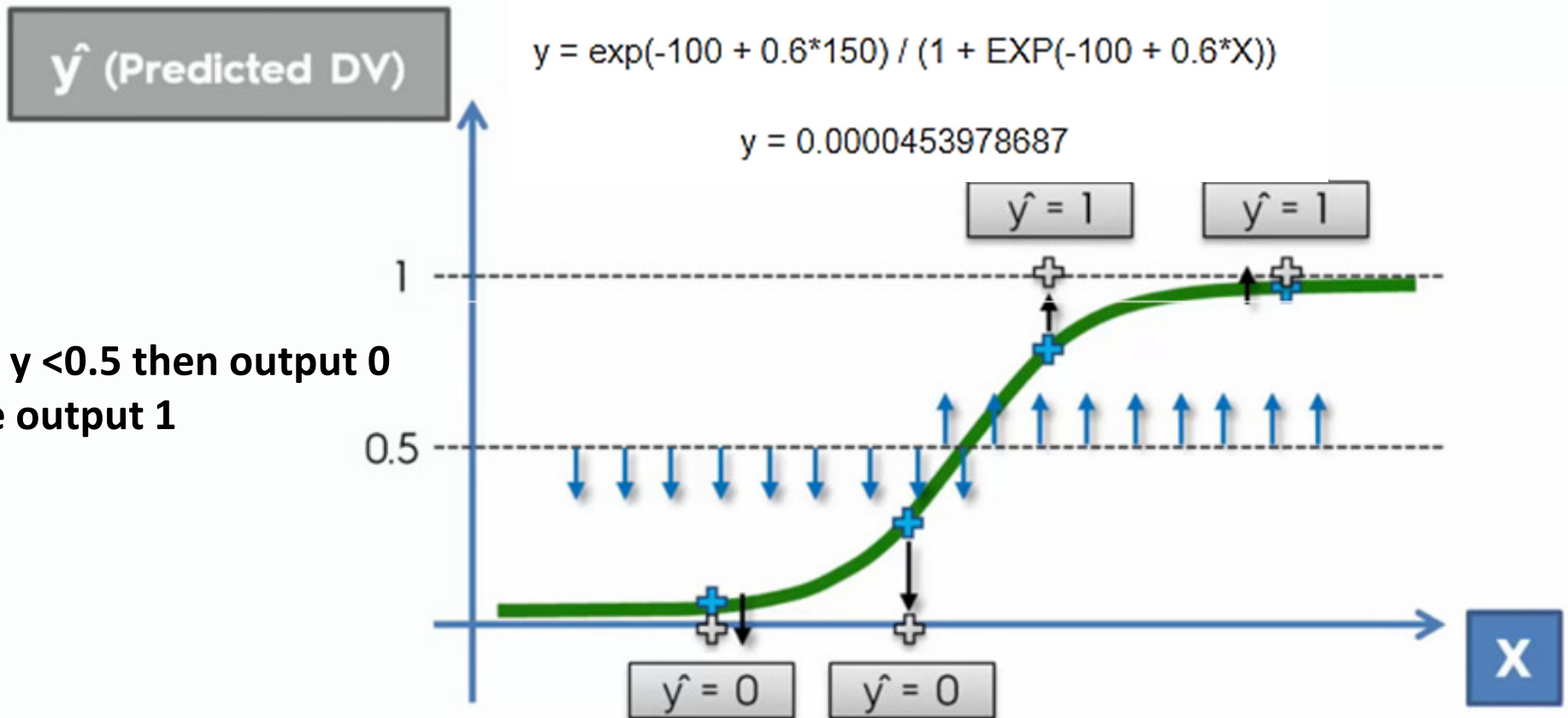
# Logistic Regression Classification

$$y = e^{(b_0 + b_1 X)} / (1 + e^{(b_0 + b_1 X)})$$

$$y = \exp(-100 + 0.6 \cdot 150) / (1 + \exp(-100 + 0.6 \cdot X))$$

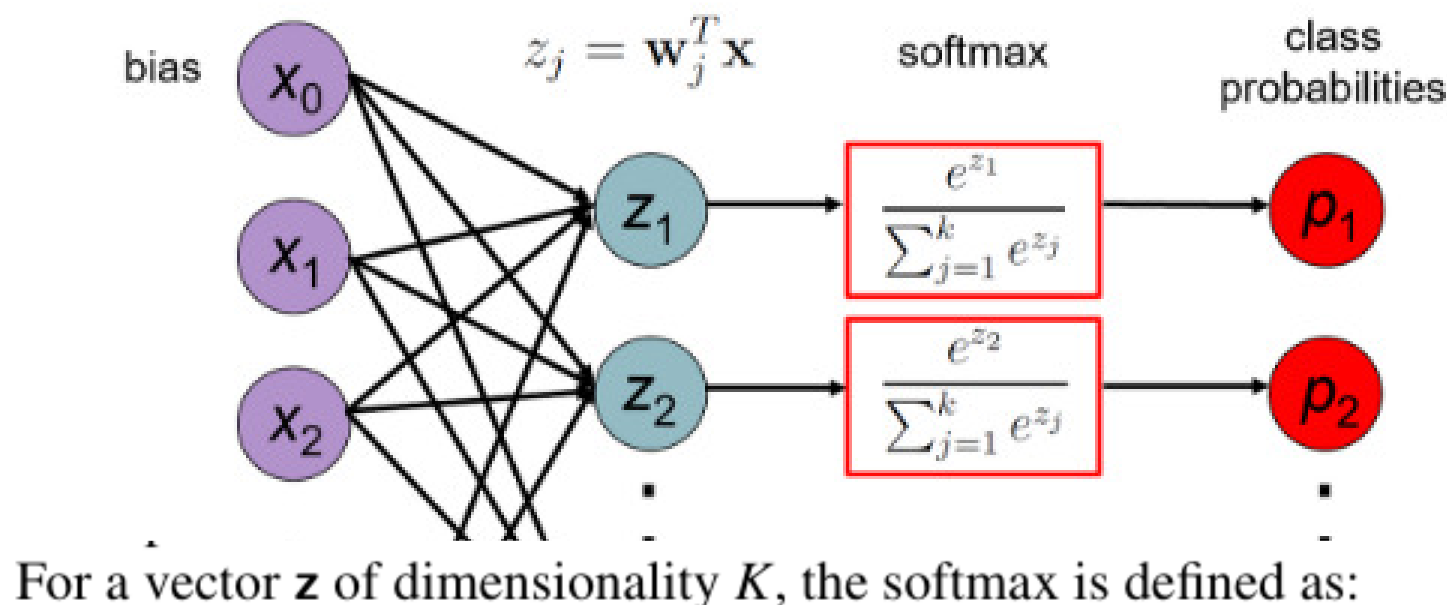
$$y = 0.0000453978687$$

If  $y < 0.5$  then output 0  
Else output 1



# Multinomial logistic regression

Sometimes we need more than two classes. Perhaps we might want to do a 3-way sentiment classification (positive, negative, or neutral).



$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)} \quad 1 \leq i \leq K$$

The softmax of an input vector  $\mathbf{z} = [z_1, z_2, \dots, z_K]$  is thus a vector itself:

$$\text{softmax}(\mathbf{z}) = \left[ \frac{\exp(z_1)}{\sum_{i=1}^K \exp(z_i)}, \frac{\exp(z_2)}{\sum_{i=1}^K \exp(z_i)}, \dots, \frac{\exp(z_K)}{\sum_{i=1}^K \exp(z_i)} \right]$$

# Learning in Logistic Regression

We need a loss function that expresses, for an observation  $x$ , how close the classifier output ( $\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$ ) is to the correct output ( $y$ , which is 0 or 1). We'll call this:

$$L(\hat{y}, y) = \text{How much } \hat{y} \text{ differs from the true } y \quad (5.20)$$

This is called **conditional maximum likelihood estimation**: we choose the parameters  **$\mathbf{w}$ ,  $b$**  that maximize the log probability of the true  **$y$  labels** in the training data given the observations  $x$ .

**The Bernoulli distribution** is the discrete probability distribution of a random variable which takes a binary, Boolean output: 1 with probability  $p$ , and 0 with probability  $(1-p)$ .




express the probability  $p(y|x)$  that our classifier produces for one observation as the following (keeping in mind that if  $y = 1$ , Eq. 5.21 simplifies to  $\hat{y}$ ; if  $y = 0$ , Eq. 5.21 simplifies to  $1 - \hat{y}$ ):

$$p(y|x) = \hat{y}^y (1 - \hat{y})^{1-y} \quad (5.21)$$

Now we take the log of both sides. This will turn out to be handy mathematically, and doesn't hurt us; whatever values maximize a probability will also maximize the log of the probability:

$$\begin{aligned} \log p(y|x) &= \log [\hat{y}^y (1 - \hat{y})^{1-y}] \\ &= y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \end{aligned} \quad (5.22)$$

Eq. 5.22 describes a log likelihood that should be maximized. In order to turn this into a loss function (something that we need to minimize), we'll just flip the sign on Eq. 5.22. The result is the cross-entropy loss  $L_{CE}$ :


$$L_{CE}(\hat{y}, y) = -\log p(y|x) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \quad (5.23)$$

Finally, we can plug in the definition of  $\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$ :

$$L_{CE}(\hat{y}, y) = -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log(1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \quad (5.24)$$

# The Gradient for Logistic Regression

It turns out that the derivative of this function for one observation vector  $x$  is Eq. 5.30 (the interested reader can see Section 5.10 for the derivation of this equation):

$$\begin{aligned}\frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial w_j} &= [\sigma(\mathbf{w} \cdot \mathbf{x} + b) - y]x_j \\ &= (\hat{y} - y)x_j\end{aligned}\tag{5.30}$$

You'll also sometimes see this equation in the equivalent form:

$$\frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial w_j} = -(y - \hat{y})x_j\tag{5.31}$$

Note in these equations that the gradient with respect to a single weight  $w_j$  represents a very intuitive value: the difference between the true  $y$  and our estimated  $\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$  for that observation, multiplied by the corresponding input value  $x_j$ .

$$\frac{\partial L_{\text{CE}}(\hat{Y} - Y)}{\partial W} = (\hat{Y} - Y)^T X$$

$$w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x; w), y)$$

# The Learning Rate-Hyperparameter

$$w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x; w), y)$$

The learning rate  $\eta$  is a **hyperparameter** that must be adjusted. If it's too high, the learner will take steps that are too large, overshooting the minimum of the loss function. If it's too low, the learner will take steps that are too small, and take too long to get to the minimum. It is common to start with a higher learning rate and then slowly decrease it, so that it is a function of the iteration  $k$  of training; the notation  $\eta_k$  can be used to mean the value of the learning rate at iteration  $k$ .