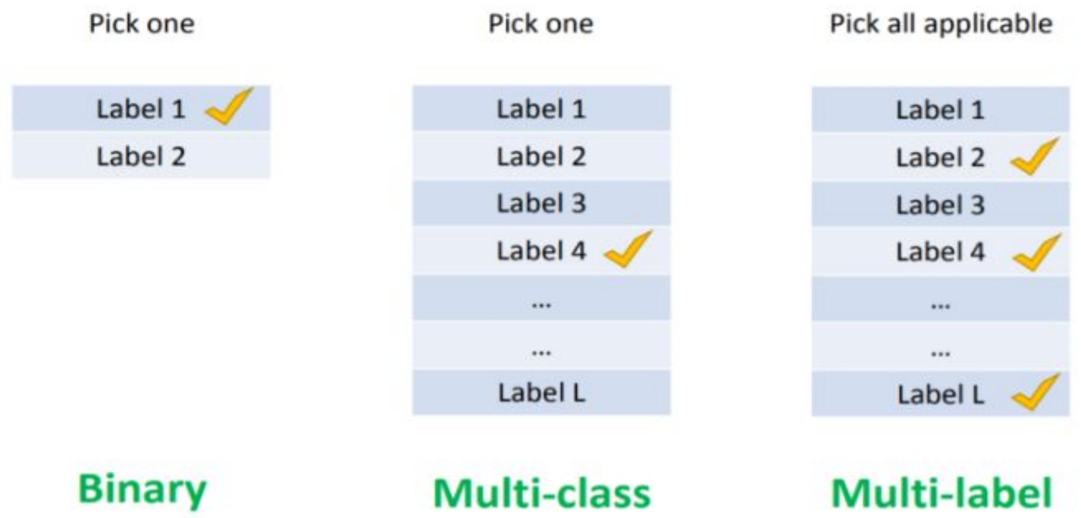
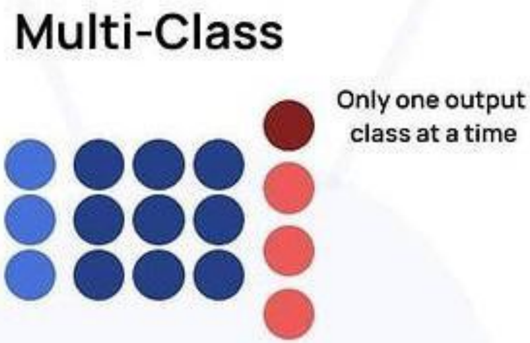
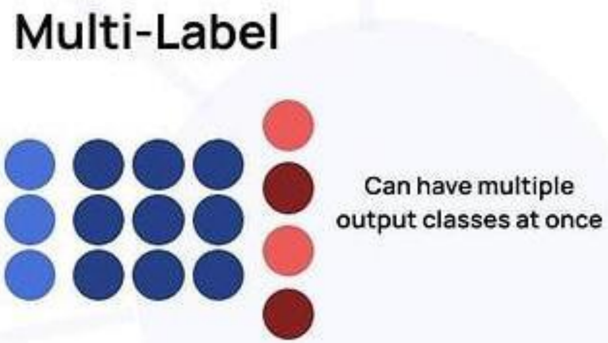
Classification

Classification



Multilabel classification

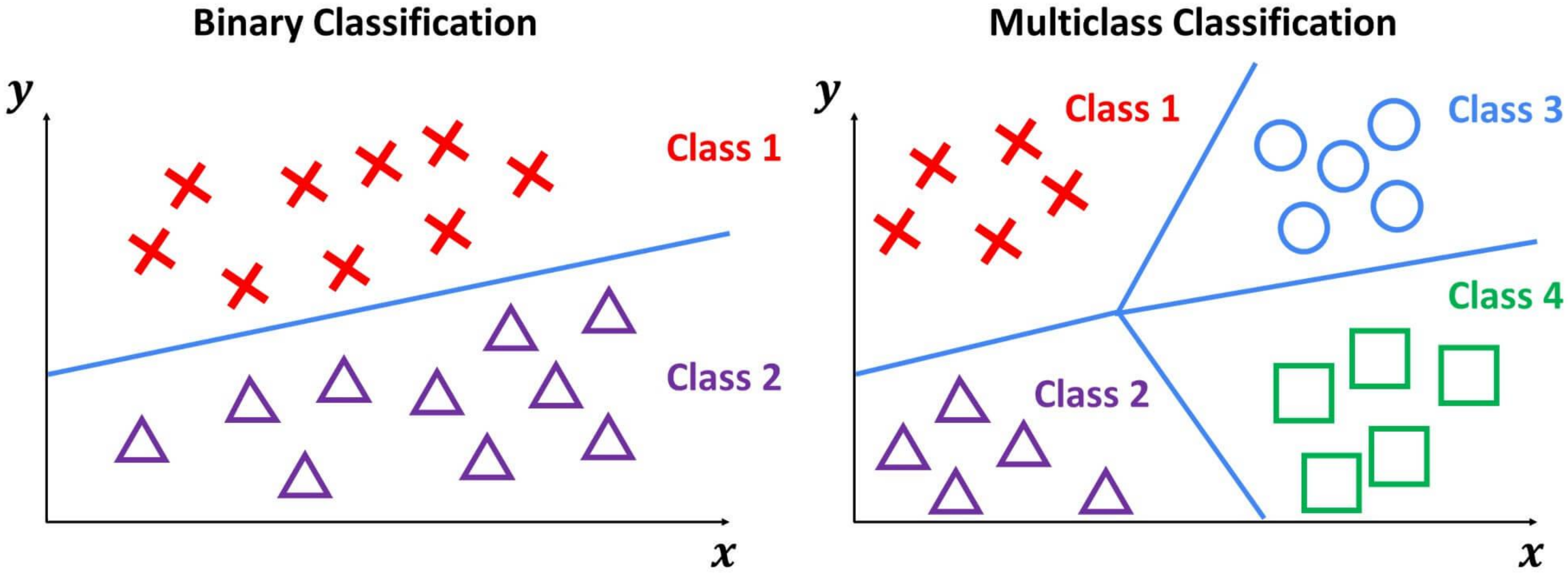




Review	sentiment
Very good quality though	Positive
The design is very odd	Negative
I advise EVERYONE DO NOT BE FOOLED!	Negative
So Far So Good!	Positive
Works great!	positive

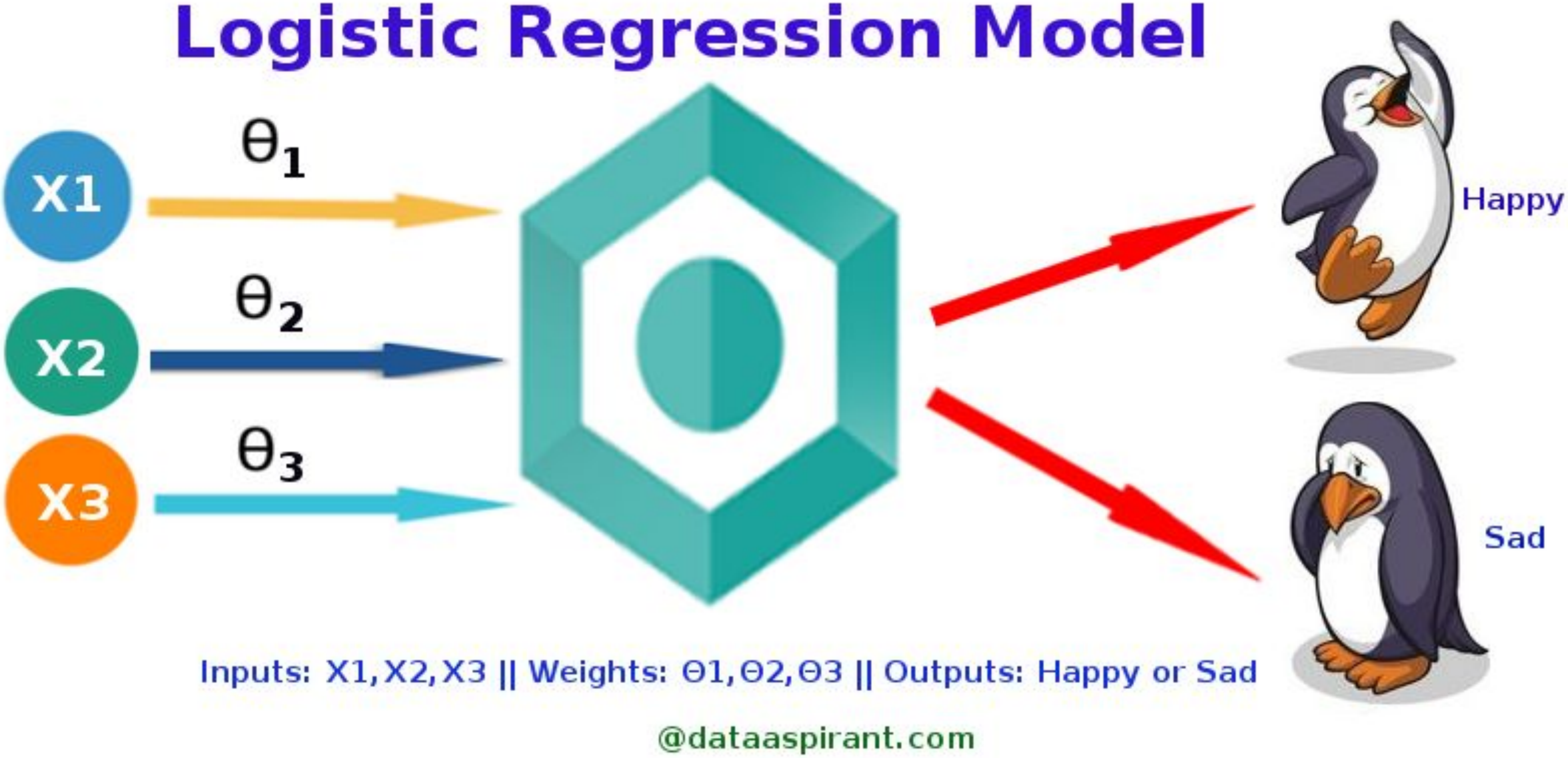
movie	Adventure	Comedy	Fantasy	Crime
Jumanji (1995)	1	0	1	0
Puccini for Beginners (2006)	0	1	0	0
How the Grinch Stole Christmas! (1966)	0	1	1	0

Multi-Class Classification

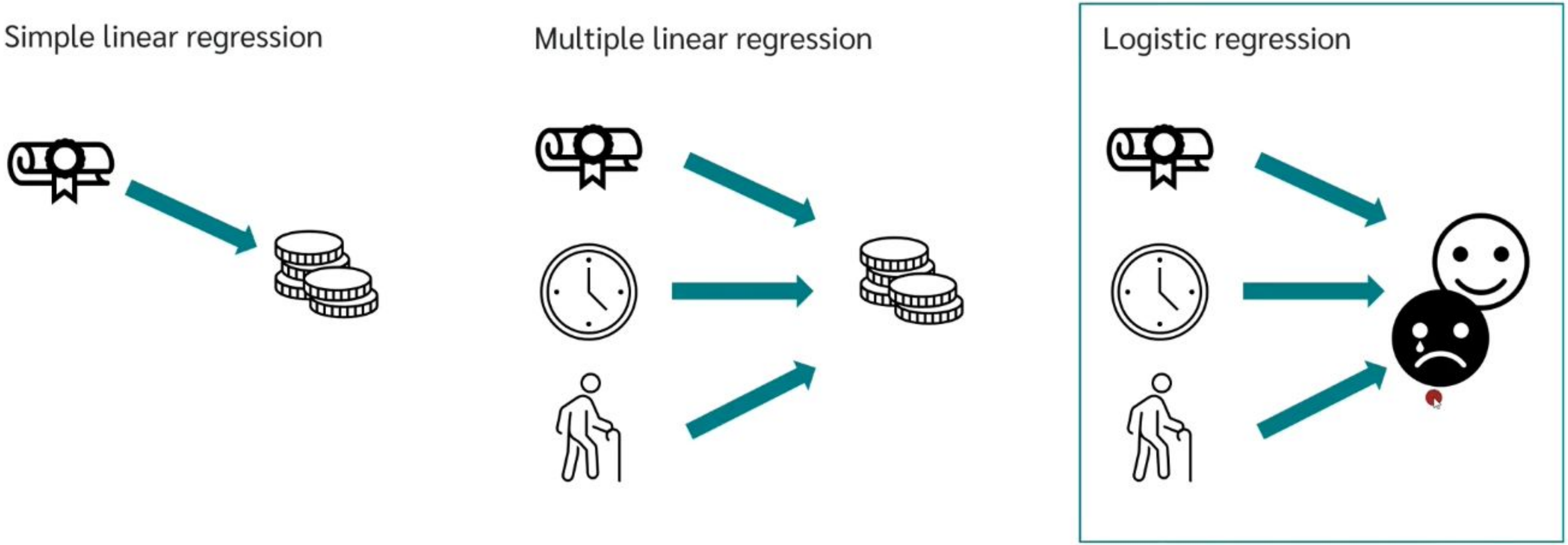


Session 3

Logistic Regression


Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1,X2,X3 || Weights: Θ1,Θ2,Θ3 || Outputs: Happy or Sad
@dataaspirant.com

What is Logistic Regression?



What is Logistic Regression?

Logistic regression is a special case of regression analysis and is calculated when the dependent variable is nominally or ordinally scaled.

Business example:

For an online retailer, you need to predict which product a particular customer is most likely to buy. For this, you receive a data set with past visitors and their purchases from the online retailer.

Medical example:

You want to investigate whether a person is susceptible to a certain disease or not. For this purpose, you receive a data set with diseased and non-diseased persons as well as other medical parameters.

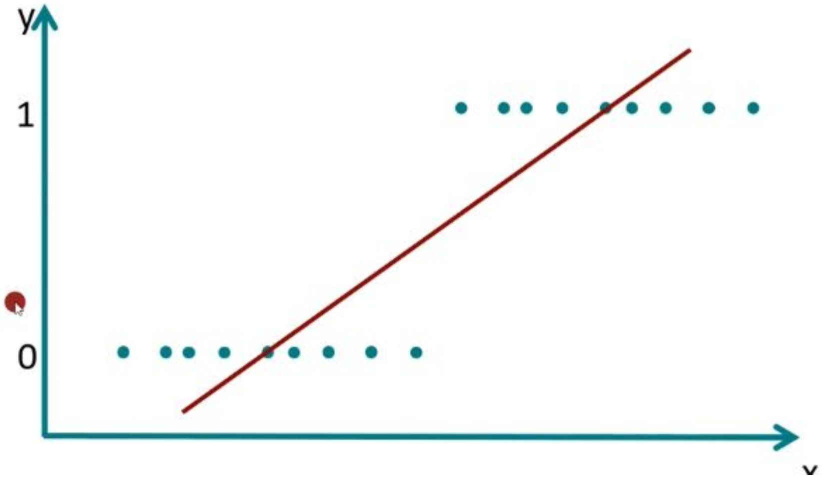
Political example:

Would a person vote for party A if there were elections next weekend?

- https://www.geeksforgeeks.org/getting-started-with-classification/
- https://www.enjoyalgorithms.com/blogs/classification-and-regression-in-machine-learning

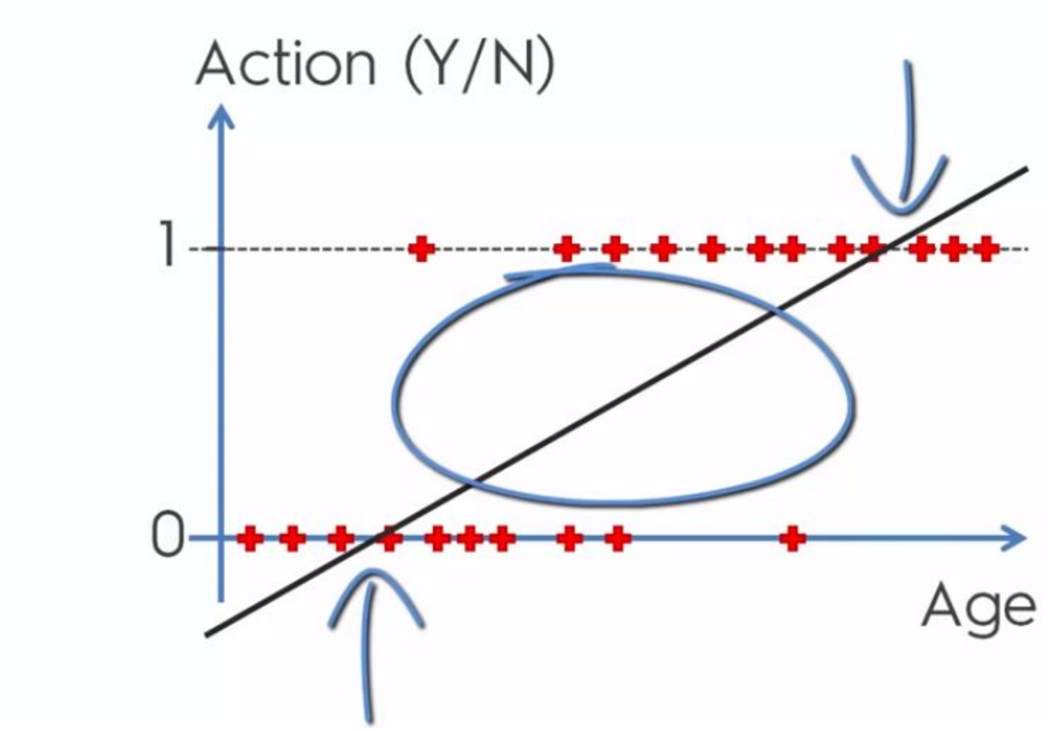
Why not just linear regression?

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

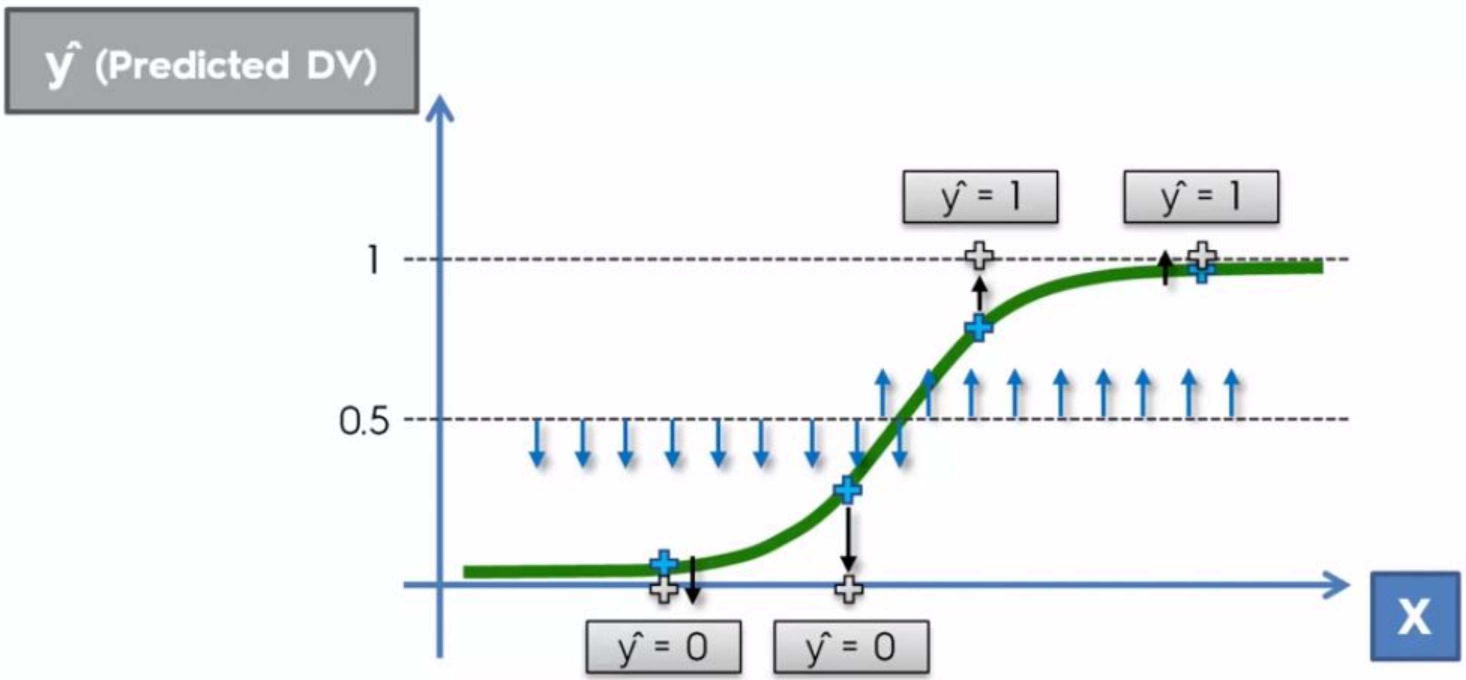


- The graph shows that values between **plus and minus infinity** can now occur.
- The goal of logistic regression is to estimate the **probability of occurrence**, not the value of the variable itself.
- The range of values for the prediction is restricted to the range between 0 and 1.
- To ensure that only values between 0 and 1 are possible, the logistic function f is used.

Logistic Regression



Logistic Regression

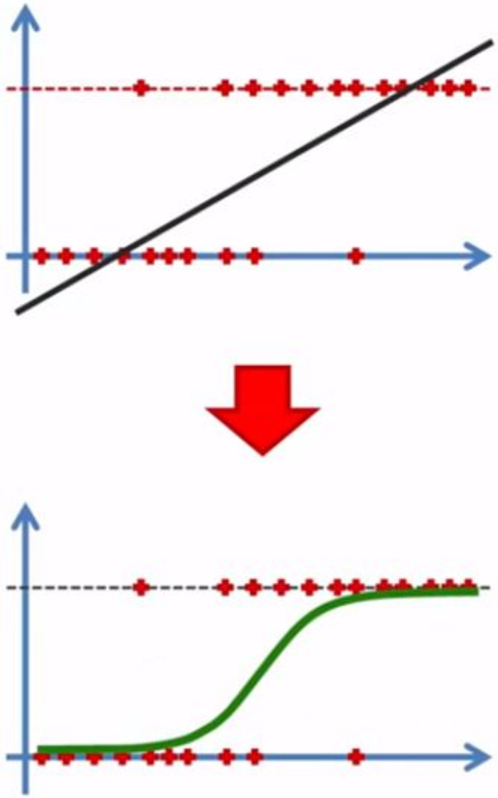


Logistic Regression

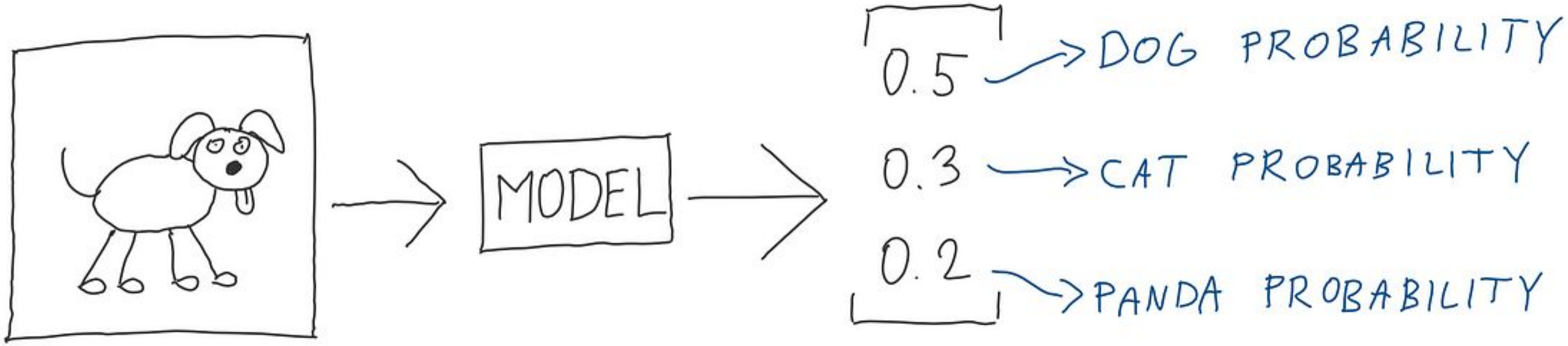
$$y = b_0 + b_1 * x$$

Sigmoid Function

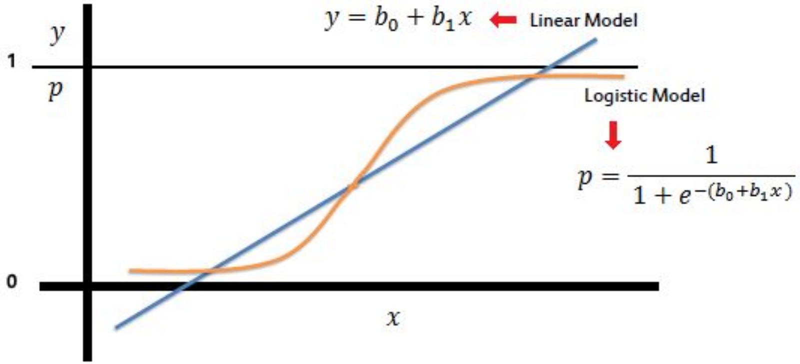
$$p = \frac{1}{1 + e^{-y}}$$



Logistic Regression



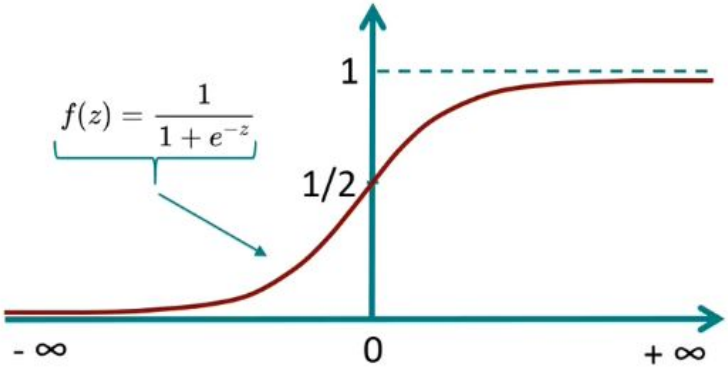
Logistic Regression



A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1

What is Logistic Regression?

The logistic model is based on the logistic function.
The important thing about the logistic function is, that only values **between 0 and 1** are possible.



$$f(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}$$

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

What is Logistic Regression?

$$z = \Theta^T x$$

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

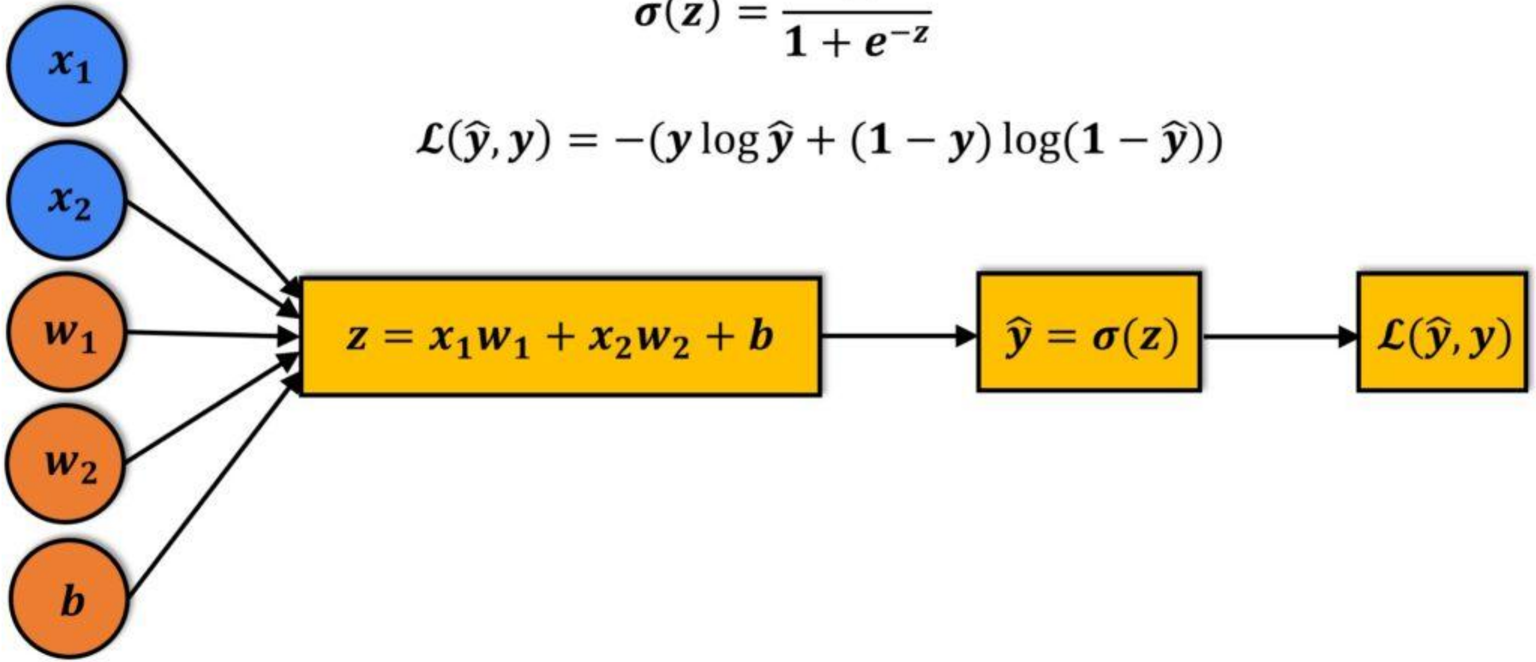
Where:

- Θ = is the weight.

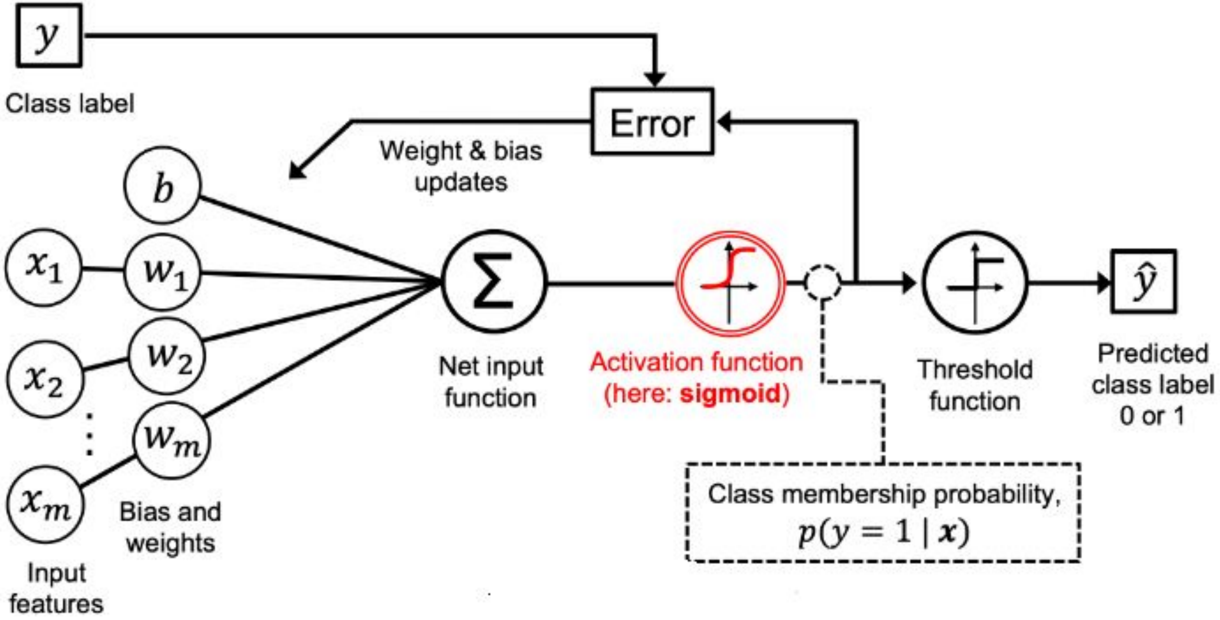
```
def sigmoid(X, weight):
    z = np.dot(X, weight)
    return 1 / (1 + np.exp(-z))
```

$$\boldsymbol{\sigma}(\boldsymbol{z}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{z}}}$$

$$\mathcal{L}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -(\boldsymbol{y} \log \hat{\boldsymbol{y}} + (\mathbf{1} - \boldsymbol{y}) \log(\mathbf{1} - \hat{\boldsymbol{y}}))$$



Logistic Regression



Cross Entropy Loss Function

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Cost function

In binary logistic regression, the cost function (also called the objective or loss function) is used to measure the error or mismatch between the predicted probabilities and the actual binary outcomes in the training data. The goal of logistic regression is to minimize the cost function, which can be accomplished by adjusting the coefficients or weights of the model.

The most commonly used cost function in binary logistic regression defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y$$

Case 1: Actual Class $y_i = 1$

- If the model gives a high probability prediction
  the value $\log(\hat{y}_i)$ will be close to 0 (since $\log(1$
- If the model gives a low probability prediction (
  the value $\log(\hat{y}_i)$ will be large and negative (sin
  very large.

Illustrative Examples

Example 1: Correct Prediction (Low Loss)

- If $y_i = 1$ and $\hat{y}_i = 0.9$:
  $\text{loss} = -\log(0.9) \approx 0.105$ (small loss)
- If $y_i = 0$ and $\hat{y}_i = 0.1$:
  $\text{loss} = -\log(0.9) \approx 0.105$ (small loss)

Example 2: Incorrect Prediction (High Loss)

- If $y_i = 1$ and $\hat{y}_i = 0.1$:
  $\text{loss} = -\log(0.1) \approx 2.302$ (large loss)
- If $y_i = 0$ and $\hat{y}_i = 0.9$:
  $\text{loss} = -\log(0.1) \approx 2.302$ (large loss)

The following is a summary of the gradient descent algorithm for logistic regression:

1. Initialize the coefficients or weights with small random values.
2. Compute the predicted probabilities h0(x^i)for each observation in the training data using the logistic function with the current coefficients or weights.
3. Compute the gradient of the cost function with respect to each coefficient or weight using the predicted probabilities and the actual binary outcomes in the training data.
4. Update the coefficients or weights using the update rule.
5. Repeat steps 2–4 until the cost function is minimized or a stopping criterion is met.

Parametric

- Fast
- Simple
- Less data

- Limited complexity
- Strong assumptions
- Poor fit (if assumptions are not correct)

Linear Regression

Logistic Regression

Nonparametric

- Flexible
- Powerful
- Effective

- More data
- Computationally expensive
- Hard to interpret if models are too complex

K-Nearest Neighbors

Decision Trees

Random Forest

```
From sk.linear_model import LogisticRegression

Model = LogisticRegression()
model.fit(x_train , y_train) # y = 0 or 1
model.predict(X_test) # give u 0 or 1
```