Logistic Regression

Logistic regression is a popular statistical method used for binary classification problems, where the goal is to predict a categorical outcome with two possible classes (e.g., yes/no, true/false, 1/0). Despite its name, logistic regression is actually a classification algorithm, not a regression algorithm.

Types of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
3. **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

considerations for using logistic regression:

1. Binary or Categorical Outcome:
    - Logistic regression is primarily used when the dependent variable is categorical.
    - It's most commonly used for binary outcomes (two classes), but can be extended to multi-class problems (multinomial logistic regression).
2. Independence of Observations:
    - The observations should be independent of each other.
    - This means that the data points should not be from repeated measurements or matched data.

3. Linearity in the Logit:
   - The log odds of the outcome should be linearly related to the independent variables.
   - This assumption can be checked using scatter plots or by adding interaction terms.

4. Absence of Multicollinearity:
   - The independent variables should not be highly correlated with each other.
   - High multicollinearity can lead to unstable estimates and large standard errors.
   - Can be checked using correlation matrices or Variance Inflation Factor (VIF).

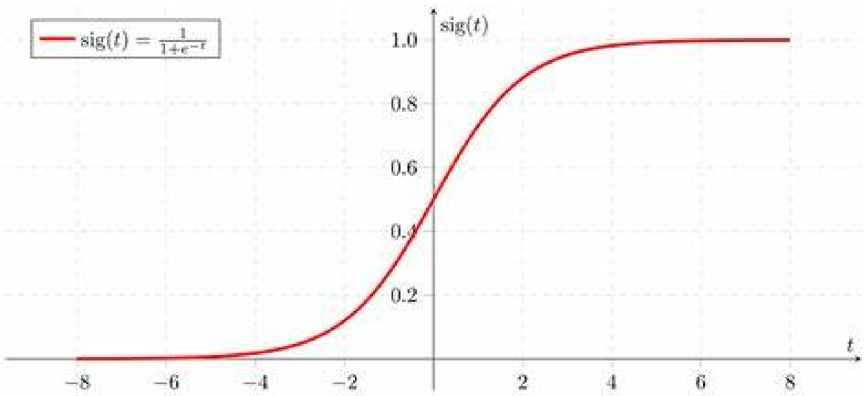
5. Absence of Outliers:
   - Logistic regression is sensitive to outliers, which can have a large impact on the results.
   - Outliers should be investigated and potentially removed or transformed.

6. Large Sample Size:
   - Logistic regression typically requires a large sample size.
   - As a rule of thumb, it's recommended to have at least 10 events per independent variable.

Sigmoid Function

The formula of the sigmoid activation function is:

$$F(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$



Logistic Regression for Binary Classification:

Logistic regression is a statistical method used for predicting a binary outcome (i.e., a result with only two possible values, typically labeled as 0 and 1). Despite its name, logistic regression is actually used for classification rather than regression.

step-by-step explanation of the training process:

1. logistic function:

$$P(y = 1|X) = \frac{1}{1 + e^{-x}}$$

where :

$$z = b0 + b1x1 + b2x2 + ... + bn * xn$$

2. Loss Calculation:

$$J(\theta) = (1/m) * \Sigma[-y * log(h\theta(x)) - (1 - y) * log(1 - h\theta(x))]$$

3. Gradient Calculation:

- For logistic regression, the gradient of the loss with respect to each weight is:

$$\partial L/\partial wj = (p - y) * xj$$

- And for the bias:

$$\partial L/\partial b = p - y$$

4. Parameter Update:

$$wj = wj - \alpha * \partial L/\partial wj$$

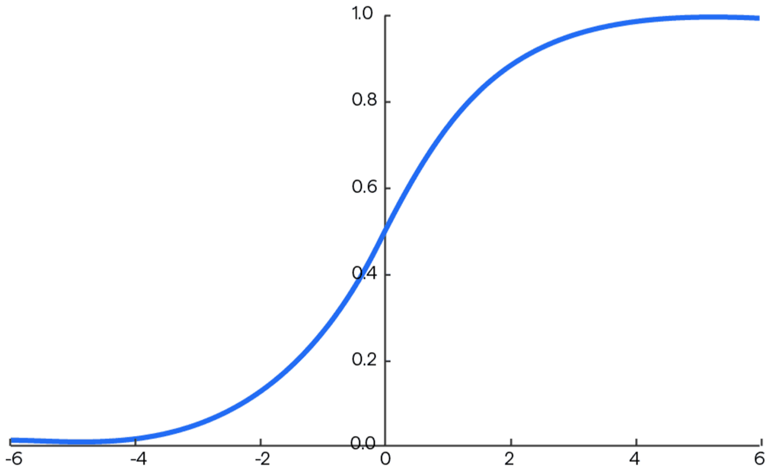
$$b = b - \alpha * \partial L/\partial b$$

  - Where α is the learning rate.

Softmax function

The formula of the Softmax activation function is:

$$softmax(z_i) = \frac{exp(z_i)}{\Sigma exp(z_j)}$$



Logistic Regression for Multiclass Classification:

For problems with more than two classes, we use multinomial logistic regression (also known as softmax regression). There are two main approaches:

1. One-vs-Rest (OvR) or One-vs-All (OvA):
    - Train a separate binary classifier for each class.
    - Each classifier distinguishes between one class and all other classes combined.
    - For prediction, run all classifiers and choose the class with the highest probability.

2. Multinomial Logistic Regression:
    - Instead of the sigmoid function, it uses the softmax function to compute probabilities for each class.
    - The softmax function ensures that the probabilities for all classes sum to 1.

step-by-step explanation of the training process:

1. logistic function:

$$P(y=1|X) = \frac{exp(z_i)}{\Sigma exp(z_j)}$$

where:

- z_i is the linear combination of features for class i
- exp is the exponential function
- The summation is over all classes

2. Loss Function: cross entropy

$$L(\theta) = -\Sigma y_k * log(h_\theta(x)_k)$$

where:

- y_k is 1 if the correct class is k, 0 otherwise
- h_θ(x)_k is the predicted probability of class k

3. Gradient Calculation

$$\partial L / \partial w_j k = (h_\theta(x)_k - y_k) * x_j$$

Where:

- w_jk is the weight for feature j of class k
- x_j is the j-th feature

$$\partial L / \partial b_k = h_\theta(x)_k - y_k$$

4. Parameter Update

The weights and biases are updated using gradient descent:

$$w_j k = w_j k - \alpha * \partial L / \partial w_j k$$

$$b_k = b_k - \alpha * \partial L / \partial b_k$$