

## **Logistic Regression**

## ❖ رگرسیون لجستیک ( Logistic Regression ) :

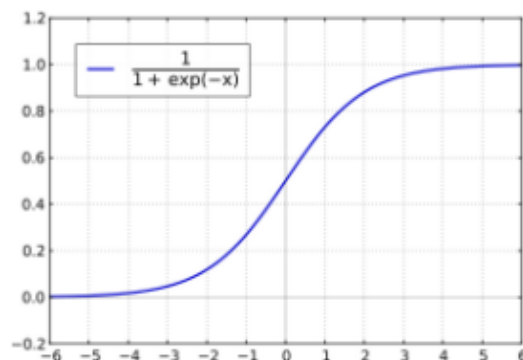
همانطور که پیش تر به آن اشاره شد برخی از الگوریتم های رگرسیون را می توان برای طبقه بندی استفاده کرد. رگرسیون لجستیک یکی از این الگوریتم ها است که به طور کلی برای تخمین احتمال متعلق بودن یک نمونه به یک کلاس خاص استفاده می شود. اگر احتمال تخمین زده شده بیشتر از ۵۰ درصد باشد ، مدل پیش بینی می کند که نمونه متعلق به آن کلاس است. به طور معمول در طبقه بندی باینری ، کلاس مثبت را ۱ و کلاس منفی را ۰ در نظر می گیرند.

### ❖ تابع فرضیه :

مدل رگرسیون لجستیک دقیقا مانند مدل رگرسیون خطی ، مقدار وزنی از ویژگی های ورودی را محاسبه می کند اما در عوض تولید نتیجه به صورت مستقیم ، لجستیک این نتیجه را تولید می کند. تابع فرضیه به صورت زیر است :

$$h_{\theta}(x) = g(\theta^T x) \quad 0 \leq h_{\theta}(x) \leq 1$$
$$g(z) = \frac{1}{1 + e^{-z}} \quad z = \theta^T x$$

به تابع  $g$  ، لجستیک یا تابع سیگموئید می گویند که خروجی آن عددی بین ۰ تا ۱ است. نمودار این تابع به صورت زیر است :



به عبارت دیگر تابع فرضیه ، احتمال آن که نمونه متعلق به کلاس ۱ باشد را نشان می دهد.

### ❖ تابع هزینه :

خروجی تابع لجستیک موج گونه است و این موضوع باعث می شود تعداد زیادی مینیمم محلی داشته باشیم به همین دلیل باید تابع هزینه متفاوتی از تابع هزینه رگرسیون خطی تعریف کنیم. داریم :

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_{\theta}(x), y) = -\log(h_{\theta}(x)) \quad \text{if } y = 1$$

$$\text{Cost}(h_{\theta}(x), y) = -\log(1 - h_{\theta}(x)) \quad \text{if } y = 0$$

اصطلاحاً به تابع بالا ، Log Loss می گویند.

می توان تابع هزینه را به صورت کلی تر نیز نوشت :

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

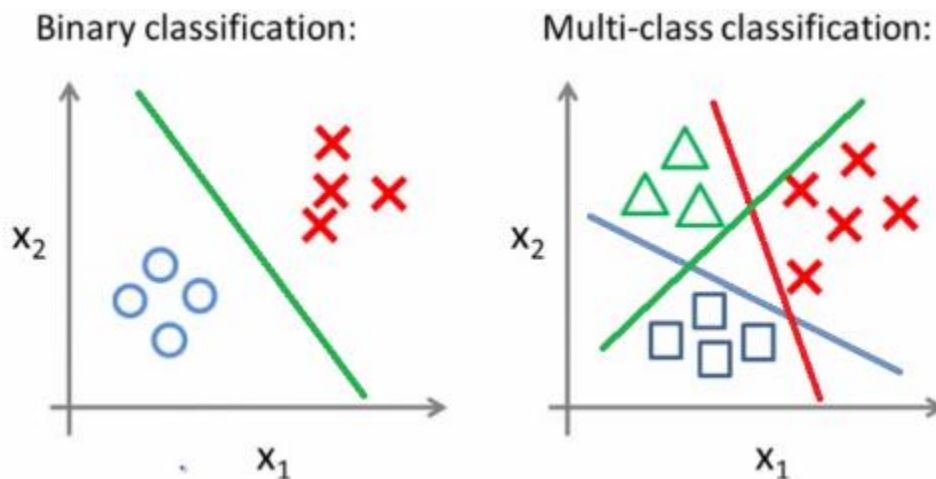
### ❖ گرادیان کاهشی در رگرسیون لجستیک :

الگوریتم گرادیان کاهشی در رگرسیون لجستیک مشابه گرادیان کاهشی در رگرسیون خطی است :

$$\begin{aligned} & \text{Repeat}\{ \\ & \theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \\ & \} \end{aligned}$$

### ❖ طبقه بندی چند کلاسه ( Multiclass Classification ) :

زمانی که به جای دو کلاس ، چند کلاس داشته باشیم ، برای طبقه بندی ، چند کلاس در واقع یک کلاس در نظر گرفته می شود و کلاس دوم نیز در مقابل آن قرار دارد. این رویه را آنقدر تکرار می کنیم تا تمامی کلاس ها به صورت مجزا با مابقی کلاس ها مقایسه شوند سپس با استفاده از تابع فرضیه ، کلاسی که بالاترین احتمال یا بالاترین مقدار را برگرداند به عنوان نتیجه خروجی در نظر می گیریم.



### ➤ پروژه iris :

می خواهیم با استفاده از دیتاست iris و با استفاده از الگوریتم رگرسیون لجستیک برنامه ای بنویسیم که ویژگی های مربوط به گل ها را به عنوان داده جدید دریافت کند و آن ها را طبقه بندی کند.

داریم :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
def load_data():
    DataSet = pd.read_csv("iris.csv", header = None,
                          names = ["sepal length",
                                   "sepal width",
                                   "petal length",
                                   "petal width",
                                   "label"])

    data = DataSet.iloc[:, :4]
    label = DataSet.iloc[:, 4]

    return data, label
```

```
data, label = load_data()

X_train, X_test, y_train, y_test = train_test_split(data, label, test_size=0.2, random_state=30)

LR = LogisticRegression()

LR.fit(X_train, y_train)

y_pred = LR.predict(X_test)

acc = accuracy_score(y_test, y_pred)

print("accuracy:", acc)
```

می توانیم نتایج این پروژه را با نتایج حاصل از مدل KNN مقایسه کنیم و مدل بهتر را انتخاب کنیم.