Makine Öğrenmesi Ders Notları
Hazırlayan: Barış Baykant Alagöz
İnönü Üniversitesi, Bilgisayar Mühendisliği, 2023-2024 Bahar Dönemi Lisans Dersi
Sorularınız ve düzeltme önerileriniz için e-mail: baykant.alagoz@inonu.edu.tr

Lojistik Regresyon ile İkili Sınıflama ve Phyton'da Uygulaması:

Lojistik regresyon çok parametreli sınıflama problemi çözümü için kullanılan bir yöntemdir.

Lineer regresyon sınıflama problemi çözümünü iyi ifade edemez. Çünkü ikili sınıflama problemleri için model çıkışı [0,1] aralığında değer vermesi istenir. Bir lineer modelin çıkışı sınırsıdır ve ikili sınıflama için gereken [0,1] aralığında sınırlı çıkışı bütün veri kümesi için üretemez. Bu nedenle çıkışı bu aralığa sınırlayan ikinci bir sınırlayıcı fonksiyona ihtiyaç duyar. Bu fonksiyon lineer(doğrusal) olmayan bir karakterdedir ve lineer modelin çıkışını bu [0,1] aralığına sıkıştırabilir.

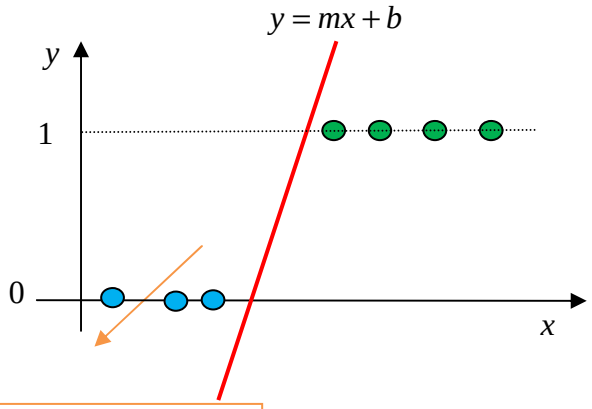
Bu bağlamda, Logistik regresyonda verileri ayıran lineer ayırıcı doğru bir sigmoid fonksiyonu yardımı ile [0,1] aralığına sınırlanır ve regresyon modelinin binary (ikili) sınıflama probleminin bir çözümünü ifade etmesi sağlanır.

Tek değişkenli durum için logistik regresyonun matematiksel geri planını inceleyelim. Aşağıda x özellik değerine karşılık verilerimiz mavi (0) ve yeşil (1) etiketileri ile ikili olarak sınıflanması istenmektedir. Örneğin, buna uygun bir eğitim kümemiz T={ $(1,0), (2.5,0), (3,0), (4.5,1), (5,1)$ $(6,1)$ } olabilir. Burada veriler (x, C) formatındadır. x özellik değeri ve $C = \{0,1\}$ ikili değer alan sınıf etiketidir.

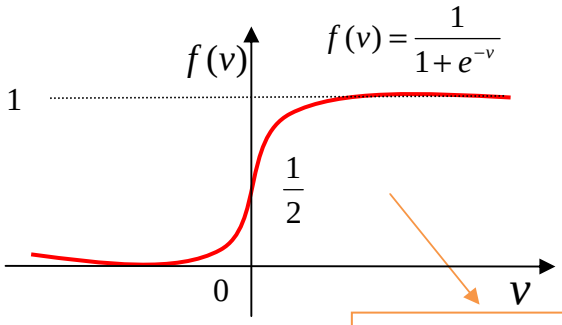
Öncelikle lineer regresyon bu probleme çözüm olabilir mi inceleyelim. Model olarak lineer bir doğru kullanılması ($y = mx + b$) istenen sonucu veremiyor çünkü lineer model çıkışı (y) sadece iki nokta da 0 ve 1 değerleri alabilir, bu iki nokta dışında ise çıkışı bir çok reel değer alabilir. Çıkışta, sınıflama için uygun olmayan çok büyük veya negatif değer verebilir. Ayrıca, bu durum gradyan iniş optimizasyon algoritması için yüksek hata değerli bir uzayda aramasına yol açar ve yakınsama performansını düşürerek eğitimi başarısız hale getirebilir. Bu sorunun çözümü için model çıkış karakteristiğini [0,1] aralığına sınırlayan bir sürekli ve türevlenebilir fonksiyona ihtiyaç vardır. Bu noktada çözüm olarak sigmoid fonksiyonu önerilmiştir.

$$f(v) = \frac{1}{1+e^{-v}}$$

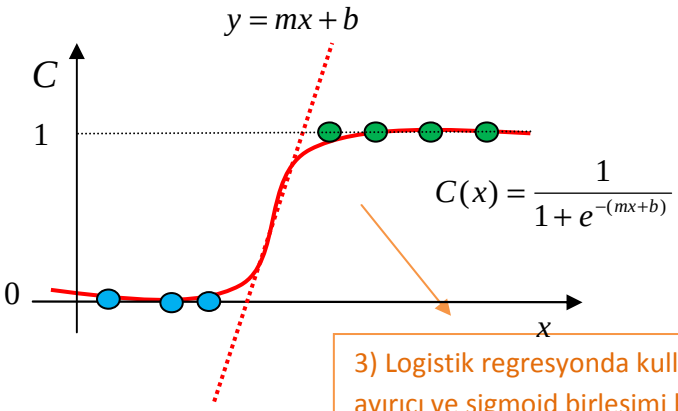
Sigmoid fonksiyonu karakteristiği aşağıdaki şekilde görüldüğü üzere [0,1] aralığında değer alan sürekli bir fonksiyondur. İki fonksiyonun birleşimi en alt da yer alan $C(x)$ ile temsil edilen sınıflama problemi çözümüne uygun fonksiyonu üretir. Bu iki fonksiyonun birleşimi ile ikili verileri temsil eden bir model elde edilir.



1) Lineer doğru ayırıcı bir fonksiyondur ve {0,1} değerlerini bir çok noktada üretemez. Bu nedenle doğrusal olamayan bir sınırlayıcı fonksiyona ihtiyaç duyulur.



2) Ayırıcı doğrunun Sigmoid fonksiyonu ile bileşkesi çıkışı [0,1] aralığında sınırlar. Aynı zamanda türevlenebilir bir fonksiyon olduğu için gradyan iniş algortiması



3) Logistik regresyonda kullanılan lineer ayırıcı ve sigmoid birleşimi karakteristiği kırmızı ile görülen forma getirir.

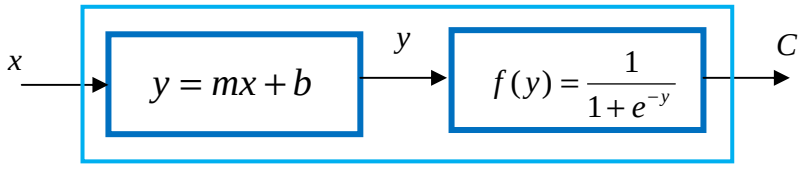
Dolayısı ile lojistik regresyon probleminde sınıflama için uygun bir fonksiyon karakteristiği aşağıdaki iki fonksiyonun bileşkesi olan $C(x) = f(mx+b)$ elde edelim.

$$y = mx + b$$

$$f(y) = \frac{1}{1+e^{-y}}$$

ve $C(x) = \frac{1}{1+e^{-mx-b}}$ ile ifade edilebilir.

Şeklinde komposit fonksiyon(bileşke fonksiyon) olarak elde edilir.



Burada ikili sınıflama problemlerine uygun çıkış veren $C(x) \in [0,1]$ model çıkışları aynı zamanda x verisinin "1" yada "0" sınıfına yakınlığını ifade eder. $C(x)$ değeri en yakın tamsayıya yuvarlanarak $\{0,1\}$ değerleri alan ikili çıkış vermesi sağlanabilir.

$$round(C(x)) \in \{0,1\}$$

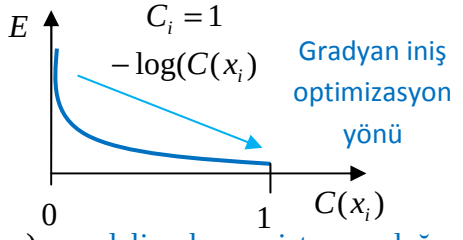
Sınıflama probleminde daha iyi performans sağlaması nedeni ile çapraz entropi kayıp fonksiyonun gradyan iniş yöntemi minimizasyonu ile çözülür. Çapraz entropi kayıp fonksiyonu sınıflama problemlerinde karesel hata kayıp fonksiyonuna göre daha avantajlı olduğu kabul edilir. Çapraz entropi kayıp fonksiyonu: (kaynak: https://developer.ibm.com/articles/implementing-logistic-regression-from-scratch-in-python/)

$$E = \frac{1}{p}\sum_{i=1}^{p}\left[-C_i \log(C(x_i)) - (1-C_i)\log(1-C(x_i)\right]$$

[Not: Çapraz entropi kayıp fonksiyonun gradyan iniş ile çalışma mekanizmasını inceleyelim:

İstenen sınıf $C_i = 1$ için hata fonksiyonu

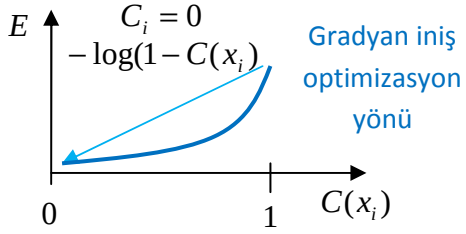
$E = \frac{1}{p}\sum_{i=1}^{p}\left[-\log(C(x_i))\right]$ indigenir.



Yandaki grafiğe göre gradyan iniş çözümü $C(x_i)$ modeli çıkışını istenen değer olan 1 değerine götürür. Çıkış doğru sınıflama sağlar.

İstenen sınıf $C_i = 0$ için hata fonksiyonu

$E = \frac{1}{p}\sum_{i=1}^{p}\left[-\log(1-C(x_i))\right]$ indigenir.



Yandaki grafiğe göre gradyan iniş çözümü $C(x_i)$ modeli çıkışını istenen değer olan 0 değerine götürür. Çıkış doğru sınıflama sağlar.]

Gradyan iniş güncelleme ifadesi çözümünü yazalım

$$m \leftarrow m - \eta\frac{dE}{dm} = m - \eta\left(\sum_{i=1}^{p}(C(x_i) - C_i)x_i\right)$$

$$b \leftarrow b - \eta\frac{dE}{db} = b - \eta\left(\sum_{i=1}^{p}(C(x_i) - C_i)_i\right)$$

Aşağıda bir x özellik değerine göre ürün kalitesi tespiti yapan uygulama yazılmıştır.

Eğitim kümesi:

Özellik değerleri

X= [32.50234527, 53.42680403, 61.53035803, 73.47563963, 86.81320787, 90.14218841, 94.21179669]

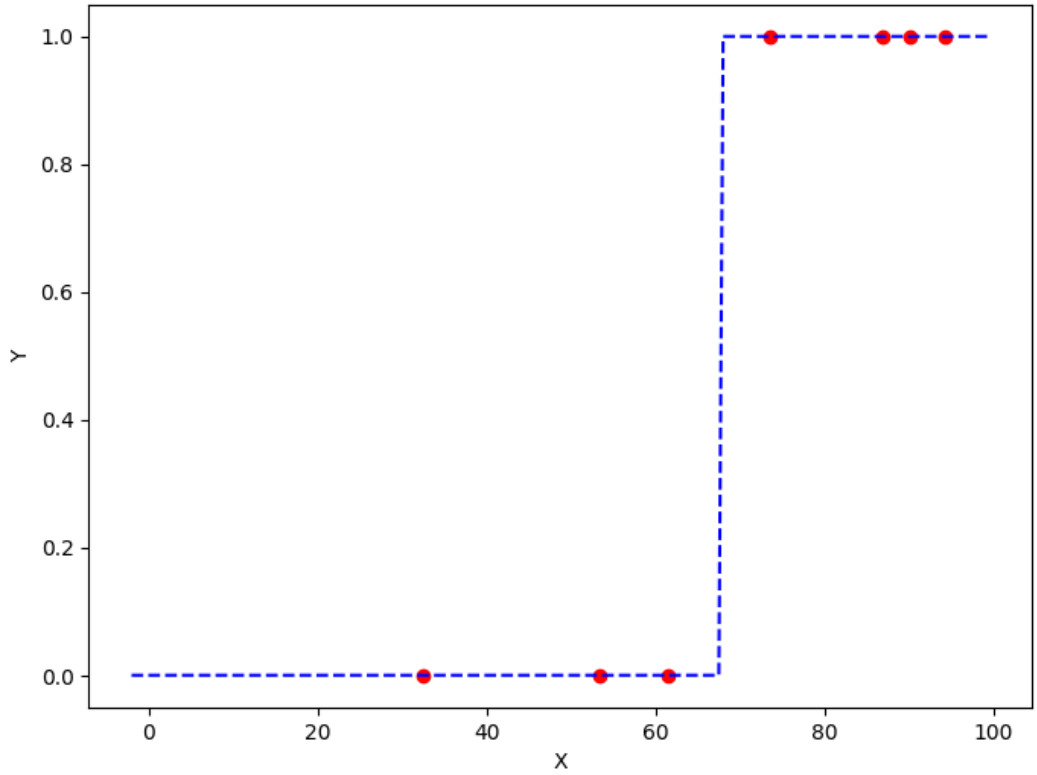
Kalitesi:

C=[0, 0, 0 , 1, 1, 1, 1]

Python kodu:

```
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
# Ölçülen X özelliğine göre ürün kalitesiz (0) veya kaliteli (1
) olarak etiketlenmiştir.
X = np.array([32.50234527, 53.42680403, 61.53035803, 73.4756396
3, 86.81320787, 90.14218841, 94.21179669]).reshape(-1,1)
Y = np.array([0, 0, 0 , 1, 1, 1, 1])

logr = linear_model.LogisticRegression()
logr.fit(X,Y)

#X_test = np.array([20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70,
 75, 80, 85, 80, 95, 100]).reshape(-1,1)
X_test = np.arange(-2,100,0.5).reshape(-1,1)
#print(X_test)
Y_pred = logr.predict(X_test)


plt.figure(figsize = (8,6))
plt.scatter(X, Y, marker='o', color='red')
plt.plot(X_test, Y_pred, color='blue',markerfacecolor='red',
    markersize=10,linestyle='dashed')
plt.xlabel("X")
plt.ylabel("Y")
plt.show()

# X=52.3 ölçülürse kalitelimidir?
X_predict=np.array([34.5, 62,1, 70.2, 89.6]).reshape(-1,1)
predicted = logr.predict(X_predict)
print(X_predict)
print(predicted)
```

Elde edilen sonuçlar:




```
[[34.5]
 [62. ]
 [ 1. ]
 [70.2]
 [89.6]]
[0 0 0 1 1]
```