Lojistik Regresyon ile İkili Sınıflandırma

1st Filiz YALÇİN
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
filiz.yaln754@gmail.com

***Özet*—100 üzerinden değerlendirilen 2 sınav sonucu bilgisi kullanılarak Lojistik Regresyon ve Stokastik Gradyan İnişi (Stochastic Gradient Descent) yöntemleri kullanılarak model eğitilmiştir. Cross Entropy Loss yöntemi ile loss (kayıp) hesabı yapılmış, parametrelerle oynanarak modelin başarısı üzerindeki etkileri gözlemlenmiştir.**

Anahtar Kelimeler— Machine Learning, Logistic Regression, Stochastic Gradient Descent, Cross Entropy Loss, Sigmoid Function

I. GIRIŞ

Çalışmanın Kapsamı: Bu çalışma, Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü Yüksek Lisans programında Makine Öğrenmesi (BLM5110) dersi kapsamında gerçekleştirilmiştir. Tüm işlemler Python dilinde programlanmıştır. Tüm işlemlerle ilgili koda ulaşmak için bkz. GitHub projesi.

Problem: Bir firmaya iş başvurusunda bulunan kişilere yapılan mülakatta iki sınav sonucuna göre iş alımları yapılmaktadır. 100 kişinin iki sınav sonucu ve işe alınıp alınmadığı bilgilerini içeren bir veri seti D<X, Y> (X: x1, x2]) bulunmaktadır. Buna göre sınav notları verilen bir kişinin işe kabul edilip edilmeyeceğini bulan algoritma hazırlanmalıdır.

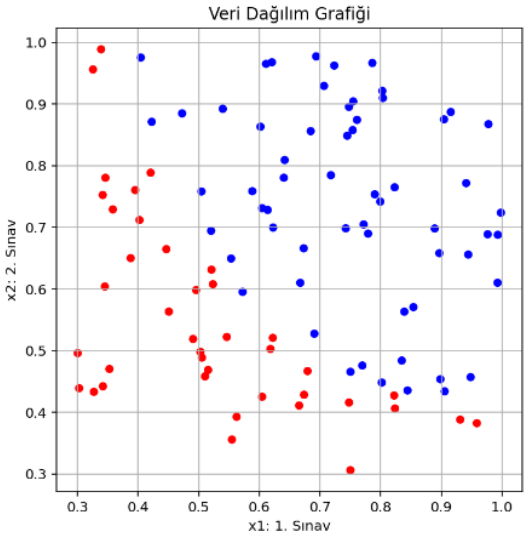
Çalışmanın Amacı: Lojistik Regresyon yönteminin çalışma prensibinin anlaşılması, Stochastic Gradient Descent (SGD) yönteminin çalışma prensibinin anlaşılması, hiperparametrelerin eğitim sonucu oluşan model üzerindeki etkisinin gözlemlenmesi, loss hesabı ve aktivasyon fonksiyonunun kullanımının öğrenilmesi.

II. DENEYSEL ANALIZ

A. Veri Setinin Hazırlanması

D<X, Y> veri setinin tamamının girdi değerleri (sınav sonuçları) normalize edilmiştir. Normalizasyon işleminde girdiler 100 üzerinden değerlendirilen sınavlar olduğundan tüm girdiler 100'e bölünmüştür. 1. sınav sonucu (x1) x eksenini, 2. sınav sonucu (x2) y eksenini temsil edecek şekilde ve etiket değerleri (y) 1 (kabul) ise mavi, 0 (ret) ise kırmızı olacak şekilde veriler görselleştirilmiştir. (Şekil 1)

Elde edilen verilerden ilk %60'ı eğitim, sonraki %20'si validasyon, kalan %20'si ise test için ayrılmıştır.



Şekil 1: Veri Dağılım Grafiği

B. Eğitim

Eğitimde ağırlık güncelleme için Stochastic Gradient Descent yöntemi uygulanmıştır. Bu yöntemde kullanılan algoritma aşağıdaki gibidir.

Loop {
    $for\ i = 1\ to\ m,$
    {
        $\theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right)x_j{}^{(i)}$ (for every j).
    }
}

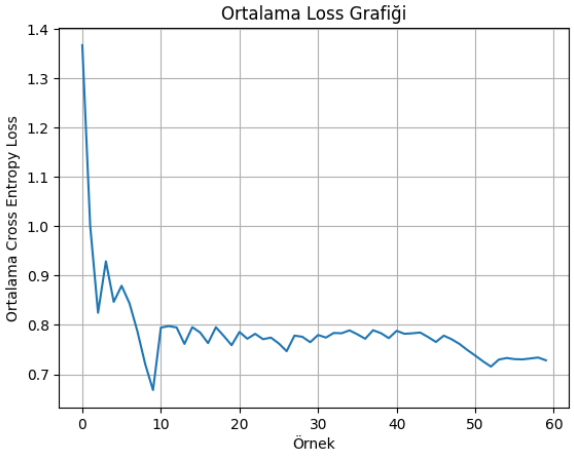
Yukarıda verilen algoritmada j, ağırlık katsayısı indisini belirtmektedir. Ağırlıklar $w_0, w_1, w_2, \dots, w_N$ şeklinde tanımlanıp w_0 bias değerini ifade edecek şekilde işlem yapılmıştır. Bias için x değeri 1 kabul edilmiştir.

Eğitim setinde bulunan her örnek için bu algoritma çalıştırılır, katsayılar o örneğe göre güncellenir. Bu yöntem Stochastic Gradient Descent (SGD) olarak isimlendirilir. [1] Bu algoritmada bulunan öğrenme katsayısı (α) 0.8 ve 0.5 olmak üzere 2 kez işlem yapılmış ve grafikler karşılaştırılmıştır.

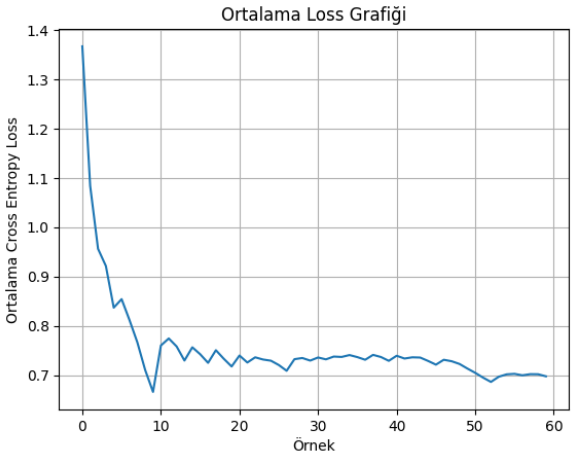
Eğitim süreci boyunca her bir örnek için loss hesabı yapılıp, ortalama loss değerleri hesaplanmıştır. Loss hesabı için Cross Entropy Loss yöntemi kullanılmıştır. Bu yöntemde kullanılan algoritma aşağıdaki gibidir.

$$\ell = -[y_{target}\log(y_{predicted}) + (1 - y_{target})\log(1 - y_{predicted})]$$

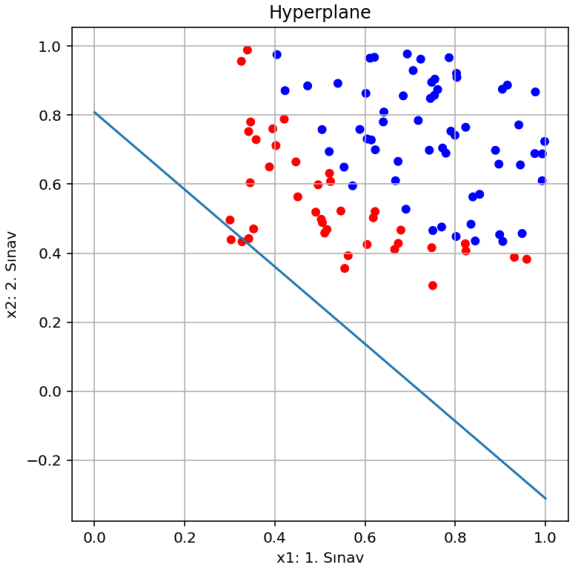
Öncelikle 1 epok ile eğitim gerçekleştirilmiş ve her bir örnek için ortalama loss hesabı yapılmıştır. Şekil 2'de görüldüğü üzere loss başlangıç noktasına göre azalmış, ancak stabil bir noktaya gelememiştir. Bundan dolayı epok sayısının arttırılması gerektiği düşünülmüştür.



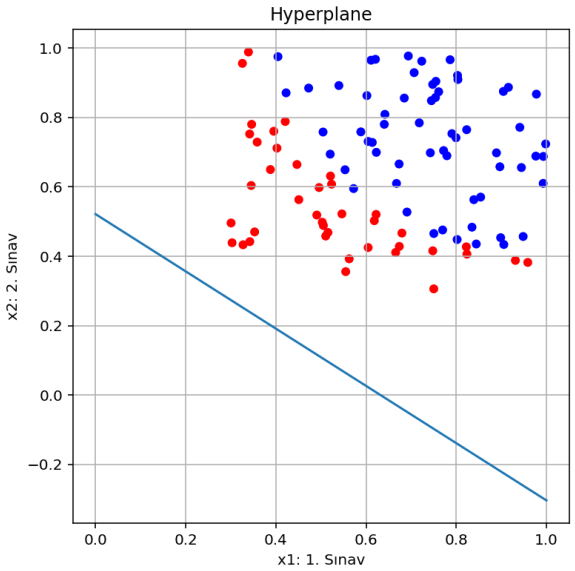
Şekil 2: 1 epok için ortalama loss grafiği (öğrenme katsayısı: 0.8)



Şekil 3: 1 epok için ortalama loss grafiği (öğrenme katsayısı: 0.5)

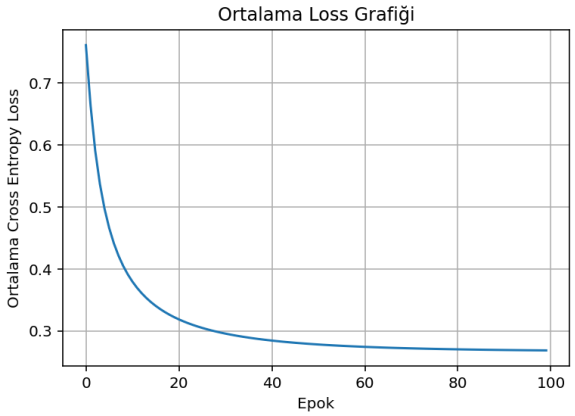


Şekil 4: 1 epok ile eğitilmiş modelin hyperplane grafiği (öğrenme katsayısı: 0.8)



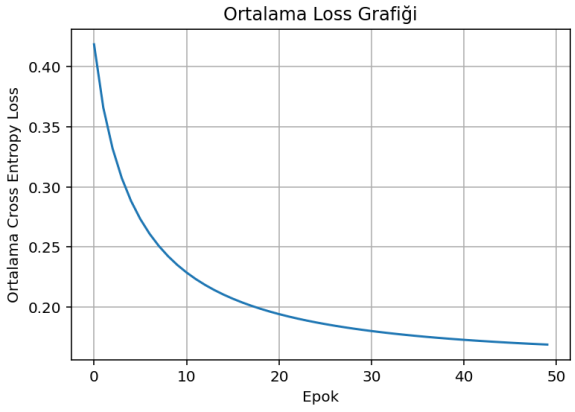
Şekil 5: 1 epok ile eğitilmiş modelin hyperplane grafiği (öğrenme katsayısı: 0.5)

Epok sayısı 100 yapıldıktan sonra her epokta bir kez loss hesabı yapılmış, ortalama loss grafiği bu sefer epok sayısına göre oluşturulmuştur.

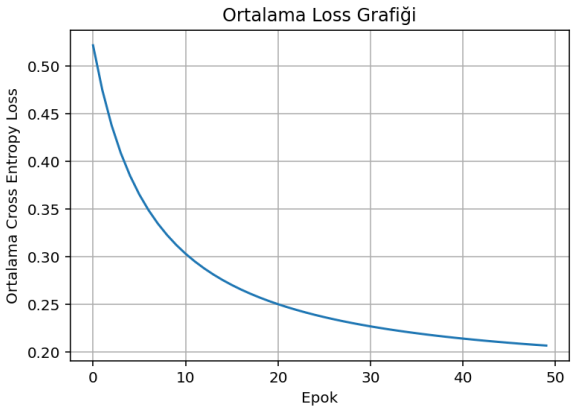


Şekil 6: 100 epok için ortalama loss grafiği

Şekil 6'da görüldüğü üzere, oluşturulan loss grafiğinde 50 epoktan sonra modelin loss'unun azalmasında yavaşlama gözlemlenmiştir. Bundan dolayı nihai model 50 epok ile oluşturulmuştur.



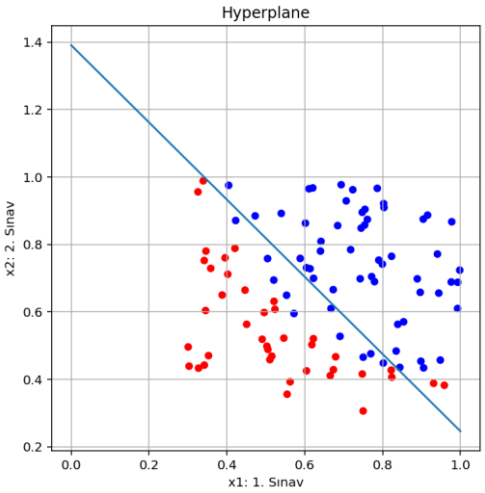
Şekil 7: 50 epok için ortalama loss grafiği (öğrenme katsayısı: 0.8)



Şekil 8: 50 epok için ortalama loss grafiği (öğrenme katsayısı: 0.5)

Şekil 7 ve Şekil 8 karşılaştırıldığında öğrenme katsayısı için 0.8 değeri kullanıldığında loss'un daha az olduğu sonucuna varılmış ve nihai model oluşturulurken bu değer kullanılmıştır.

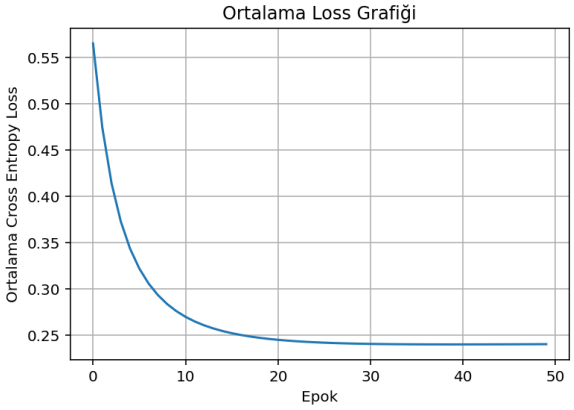
Şekil 9'da, oluşan modelin ağırlıkları kullanılarak modele ait hyperplane çizgisi (karar çizgisi) tüm veri seti ile birlikte görselleştirilmiştir.



Şekil 9: 50 epok ile eğitilmiş modelin hyperplane grafiği (öğrenme katsayısı: 0.8)

C. *Validasyon*

Eğitim sırasında her epokta elde edilen model ağırlıkları (ve bias değeri) kullanılarak validasyon için ayrılmış veri setindeki her bir örnek için loss hesaplanıp ortalama loss hesaplanmış, Şekil 10'da görselleştirilmiştir.

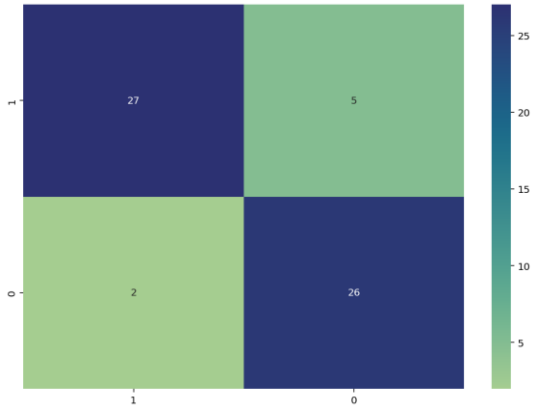


Şekil 10: Validasyon örneklerine göre ortalama loss değerleri

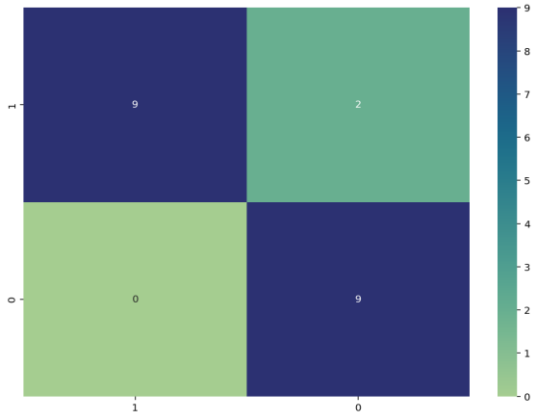
Şekil 10'da görüldüğü üzere validasyon verileri için de loss azalmaktadır. Yani aşırı öğrenme (overfit) durumu gözlenmemiştir. Bundan dolayı herhangi bir iyileştirme yapılmamıştır.

D. *Test*

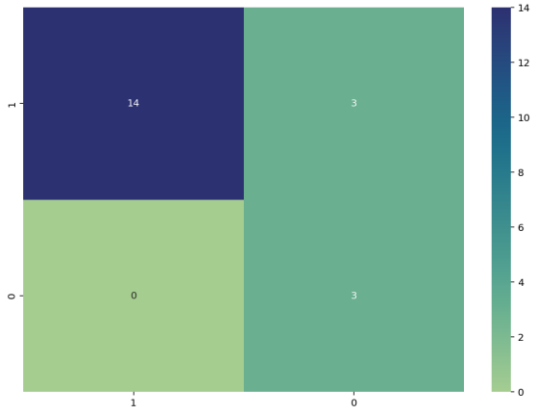
Oluşturulan model ile eğitim, validasyon ve test verileri üzerinde tahmin işlemi yapılmıştır. Modelin ağırlıkları ile karşılık geldikleri x değerleri çarpılıp toplanmış, bias değeri ise 1 ile çarpılarak sonuca eklenerek bir değer elde edilmiştir. Elde edilen değer Sigmoid fonksiyonundan geçirilerek fonksiyon sonucu 0.5'ten büyük veya eşitse 1 (Kabul), 0.5'ten küçükse 0 (Ret) olarak tahmin işlemi gerçekleştirilmiştir. Her bir veri seti için karmaşıklık matrisleri (confusion matrix) hazırlanmıştır. (Şekil 11, Şekil 12, Şekil 13)



Şekil 11: Eğitim veri seti karmaşıklık matrisi



Şekil 12: Validasyon veri seti karmaşıklık matrisi



Şekil 13: Test veri seti karmaşıklık matrisi

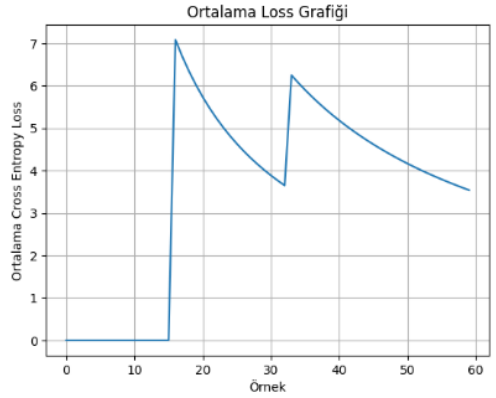
Karmaşıklık matrislerindeki bilgilere göre doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 Score metrikleri Tablo 1'de bulunmaktadır.

Veri Seti	*Doğruluk (Accuracy)*	*Kesinlik (Precision)*	*Duyarlılık (Recall)*	*F1 Score*
Eğitim	$0.88\overline{3}$	0.93103448276	0.84375	0.88524590164
Validasyon	0.9	1.0	$0.\overline{81}$	0.9
Test	0.85	1.0	0.823529412	0.90322580645

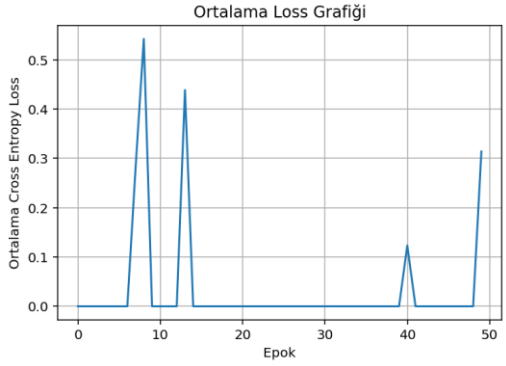
Tablo 1: Başarı metriklerinin eğitim, validasyon ve test verilerinde değerleri

III. SONUÇ

Veri üzerinde normalizasyon yapılmadan eğitim gerçekleştirildiğinde ortalama loss fonksiyonu grafiği Şekil 14 ve Şekil 15'te bulunmaktadır. Normalizasyon yapılmadığında ortaya çıkan ortalama loss grafiği incelendiğinde istenen sonucu vermediği gözlemlenmiştir. Bundan dolayı veriler normalize edildikten sonra eğitim gerçekleştirilmiştir.



Şekil 14: Normalizasyon yapılmamış veri ile eğitimde ortalama loss grafiği (1 epok)



Şekil 15: Normalizasyon yapılmamış veri ile eğitimde ortalama loss grafiği (50 epok)

1 epok ile eğitim yapılırken öğrenme katsayısı 0.8 (Şekil 2) olduğunda loss ilk adımda çok hızlı şekilde düşüş yaşamış, ancak bitiş noktasında (60. örnekte) öğrenme katsayısı 0.5 (Şekil 3) olma durumu ile kıyas edildiğinde daha yüksek bir loss ile bitmiştir. 50 epok ile eğitim yapıldığında ise Şekil 7 ve Şekil 8'de görüldüğü üzere öğrenme katsayısı 0.8 olduğunda daha başarılı bir sonuç elde edilmiştir. Bundan dolayı öğrenme katsayısı 0.8 olarak belirlenmiştir.

Şekil 4 ve Şekil 5 incelendiğinde, 1 epok ile eğitim yapıldığında oluşan karar çizgisinin eğiminin (ağırlıkların) olması gerekenden uzak olmadığı gözlemlenmiştir. Ancak bias (w_0) değeri olması gerekenden uzak kaldığı için doğru, olması gereken noktadan ötelenmiş gibi görünmektedir; dolayısıyla başarısız olmaktadır. Epok sayısı artınca bias değerinin de olması gereken noktaya geldiği gözlemlenmiştir.

REFERANSLAR

[1] A. Ng, "CS229: Machine Learning — Lecture Notes," Stanford University, 2023. [Online]. Ulaşım adresi: https://cs229.stanford.edu/main_notes.pdf. Ulaşma zamanı: 25 Kasım 2025.