Lojistik Regresyon ile İşe Alım Kararlarının Tahmini: Stochastic Gradient Descent Yöntemi

Altay Mirzaliyev
Siber Güvenlik ve Kriptografi
Yıldız Teknik Üniversitesi
Email: altay.mirzaliyev@std.yildiz.edu.tr

***Abstract*—Bu çalışmada, bir firmaya iş başvurusunda bulunan kişilerin sınav sonuçlarına göre işe kabul edilip edilmeyeceğini lojistik regresyon yöntemiyle tahmin eden bir algoritma geliştirilmiştir. Çalışma kapsamında Stochastic Gradient Descent (SGD) yöntemi kullanılarak model eğitilmiş ve modelin doğruluk, kesinlik, geri çağırma ve F-skor değerleri hesaplanmıştır. Elde edilen sonuçlar, yöntemin bu tür problemler için etkili olduğunu göstermektedir.**

***Index Terms*—Lojistik Regresyon, Stochastic Gradient Descent, Makine Öğrenmesi, Sınıflandırma**

I. Giriş

İş başvurularının değerlendirilmesinde, adayların performansını sınav sonuçlarına ve mülakat verilerine dayalı olarak analiz etmek önemli bir yere sahiptir. Bu çalışmada, lojistik regresyon kullanılarak iki sınav sonucuna dayalı bir sınıflandırma problemi çözülmüştür. Çalışmanın amacı, sınav sonuçları verilen bir adayın işe kabul edilip edilmeyeceğini tahmin etmektir. Veriler, ilk iki sütunda iki sınav sonucunu ve son sütunda ise işe kabul durumunu içermektedir. Modelin başarısı, eğitim, doğrulama ve test verileri üzerinde değerlendirilmiştir.

Veri Seti Özellikleri

Veri seti aşağıdaki sütunları içermektedir:

- Birinci sınav notu (0-100 arası)
- İkinci sınav notu (0-100 arası)
- Başarı durumu ($y = 1$: Kabul, $y = 0$: Ret)

Veri setine ait bazı temel özet istatistikler aşağıda verilmiştir:

- Toplam öğrenci sayısı: 100
- Geçen öğrenci sayısı ($y = 1$): 60
- Kalan öğrenci sayısı ($y = 0$): 40
- Ortalama sınav notu (1. Sınav): 65.64
- Ortalama sınav notu (2. Sınav): 66.22

Şekil 1'deki dağılım grafiğinde, sınavdan geçen ($y = 1$) ve kalan ($y = 0$) öğrencilerin dağılımını göstermektedir.

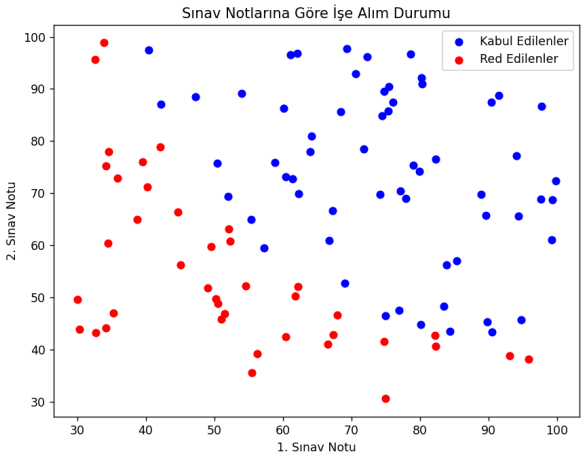

Fig. 1. Sınav Notuna Göre Kabul Durumu

II. Yöntem

Lojistik regresyon modeli, Stochastic Gradient Descent yöntemi ile eğitilmiştir. Modelin öğrenme oranı η ve iterasyon sayısı gibi hiperparametreler optimize edilmiştir. Algoritma aşağıdaki adımlarla gerçekleştirilmiştir:

- Verilerin eğitim, doğrulama ve test setlerine bölünmesi
- Lojistik regresyon fonksiyonunun tanımlanması
- SGD yöntemiyle model eğitimi
- Performans metriklerinin hesaplanması

III. Stokastik Gradyan İnişi (SGD)

A. 1)Tanım:

makine öğrenimi ve derin öğrenme modellerinde kullanılan bir optimizasyon algoritmasıdır. SGD, hata fonksiyonunu minimize etmek için model parametrelerini günceller. Batch Gradyan İnişi'nden farklı olarak, SGD her adımda yalnızca tek bir veri noktası (veya küçük bir mini-batch) kullanarak parametreleri günceller. Bu, özellikle büyük veri setleri için hesaplama açısından verimli bir yöntemdir.

$$\nabla_\theta \ell(\theta; x^{(i)}, y^{(i)}) = (h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

B. Temel Özellikleri:

- **Verimlilik**: Her adımda yalnızca bir veya birkaç veri noktası işlem görür, bu da SGD'yi büyük veri setleri için hızlı bir seçenek yapar.
- **Gürültülü Güncellemeler**: Tek bir örneğin kullanılması gürültü oluşturabilir, bu da algoritmanın yerel minimumlardan kaçmasına yardımcı olabilir ancak kararsızlığa yol açabilir.
- **Öğrenme Oranı** (η): Her adımın büyüklüğünü kontrol eden kritik bir hiperparametredir. Uygun bir öğrenme oranı seçimi yakınsama için önemlidir.
- **Yakınsama**: SGD, iterasyon başına daha hızlı yakınsar ancak gürültü nedeniyle daha fazla adım gerektirebilir. Kararlılık sağlamak için öğrenme oranı azaltma veya momentum gibi teknikler kullanılabilir.

—

C. SGD Güncelleme Formülü

SGD'nin genel güncelleme kuralı aşağıdaki gibidir:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \ell\big(\theta_t; x^{(i)}, y^{(i)}\big)$$

Burada:

- θ_t: t iterasyonundaki model parametreleri (ağırlıklar).
- η: Öğrenme oranı (adım boyutu).
- $\ell(\theta_t; x^{(i)}, y^{(i)})$: $(x^{(i)}, y^{(i)})$ veri noktası için kayıp fonksiyonu.
- $\nabla_\theta \ell(\theta_t; x^{(i)}, y^{(i)})$: Kayıp fonksiyonunun θ_t parametrelerine göre türevi (gradyanı).

—

D. Lojistik Regresyon İçin SGD (Örnek)

Lojistik regresyonda kayıp fonksiyonu çapraz entropi kaybıdır:

$$\ell(\theta; x^{(i)}, y^{(i)}) = -y^{(i)} \log(h_\theta(x^{(i)})) - (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))$$

Burada $h_\theta(x)$ sigmoid fonksiyonudur:

$$h_\theta(x) = \frac{1}{1+e^{-x}}$$

Bir veri noktası için kaybın gradyanı şu şekilde hesaplanır:

$$\nabla_\theta \ell(\theta; x^{(i)}, y^{(i)}) = \big(h_\theta(x^{(i)}) - y^{(i)}\big) x^{(i)}$$

Çapraz Entropi Hatası (Cross Entropy Loss) bu şekilde tanımlanır:

$$\mathcal{L}(\hat{y}, y) = -\left[y \log(\hat{y}) + (1-y) \log(1-\hat{y})\right] \tag{1}$$

Burada:

- $\hat{y}$: Modelin pozitif sınıf için tahmin ettiği olasılık,
- y: Gerçek etiket (pozitif sınıf için 1, negatif sınıf için 0).

—

E. SGD'nin Avantajları

- Büyük veri setleri ve çevrim içi öğrenme için uygun.
- Batch Gradyan İnişi'ne kıyasla daha hızlı iterasyonlar sağlar.
- Rastgelelik sayesinde yerel minimumlardan kaçabilir.

F. SGD'nin Dezavantajları

- Gürültü nedeniyle minimuma yakın noktalarda salınım yapabilir.
- Kararlılık sağlamak için *momentum*, *öğrenme oranı azalma* veya *mini-batch* gibi ek teknikler gerekebilir.

—

G. Uygulama Alanları

- Lojistik regresyon, doğrusal regresyon ve sinir ağları gibi makine öğrenimi modellerinin eğitilmesi.
- Derin öğrenme mimarilerinin (Örn. CNN ve RNN) optimize edilmesi.

Stokastik güncellemeler sayesinde SGD, hesaplama verimliliği ve yakınsama hızı arasında bir denge sağlar. Bu da onu modern makine öğrenimi optimizasyonunda temel bir yöntem haline getirir.

Modelin ağırlık değerlerinin SGD ile güncelledikten sonra *cost* fonksiyonunun eğrisi Şekil 2'de verilmiştir. Eğri, modelin sınıflandırma başarısını görselleştirmektedir.

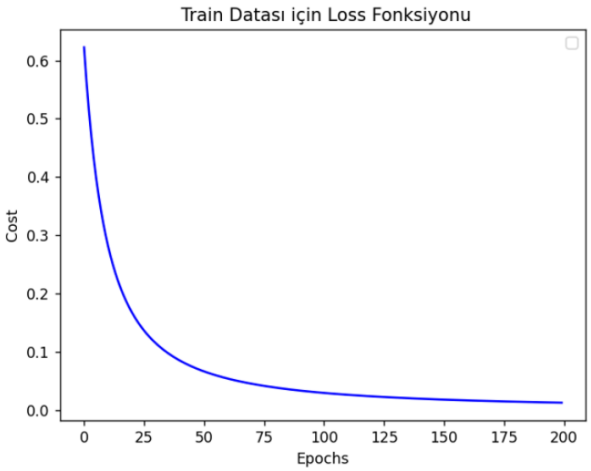

Fig. 2. Eğitim Veri Seti İçin Loss Fonksiyonu

Doğrulama veri seti ile Eğitim veri setini karşılaştıracak olursak Şekil 3'deki gibi bir fonksiyon elde ediyoruz

IV. Deneysel Analiz

Bu bölümde, modelin eğitimi sırasında elde edilen performans değerleri ve grafikler sunulmuştur. Aşağıda Tablo I performans metriklerini göstermektedir.

A. Performans Metrikleri

Tablo I, modelin eğitim, doğrulama ve test veri setlerindeki doğruluk, kesinlik, geri çağırma ve F-skor değerlerini göstermektedir.

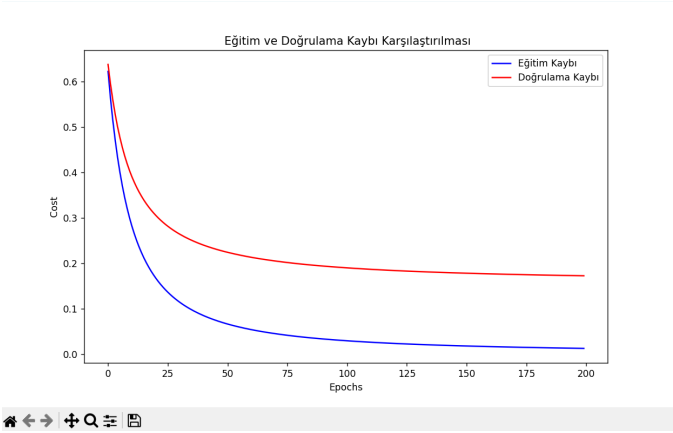

Fig. 3. Eğitim ve Doğrulama Loss Fonksiyonu karşılaştırılması

KARIŞIKLIK MATRISI (CONFUSION MATRIX)

Karışıklık matrisi, bir sınıflandırma modelinin performansını değerlendirmek için kullanılan bir araçtır. Modelin tahmin ettiği sınıflarla gerçek sınıflar arasındaki ilişkiyi görselleştirir. Bu matris, doğru ve yanlış sınıflandırmalar hakkında bilgi sağlar.

B. Karışıklık Matrisi Bileşenleri

- **True Positive (TP):** Gerçek pozitif (doğru pozitif) — Modelin pozitif sınıfı doğru bir şekilde tahmin ettiği durumlar.
- **True Negative (TN):** Gerçek negatif (doğru negatif) — Modelin negatif sınıfı doğru bir şekilde tahmin ettiği durumlar.
- **False Positive (FP):** Yanlış pozitif (tip I hata) — Modelin negatif olan bir örneği yanlış bir şekilde pozitif olarak tahmin ettiği durumlar.
- **False Negative (FN):** Yanlış negatif (tip II hata) — Modelin pozitif olan bir örneği yanlış bir şekilde negatif olarak tahmin ettiği durumlar.

Bir 2×2 karışıklık matrisi şöyle görünür:

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

C. Karışıklık Matrisi Kullanım Amacı

Karışıklık matrisi, aşağıdaki amaçlarla kullanılır:

- Modelin başarısını daha detaylı değerlendirmek: Hangi tür hataların yapıldığını anlamak (yanlış pozitif veya yanlış negatif), modelin geliştirilmesinde faydalıdır.
- Farklı metrikler hesaplamak: Doğruluk, hassasiyet (precision), duyarlılık (recall), F1 skoru gibi önemli değerlendirme metrikleri confusion matrix üzerinden hesaplanabilir.

D. Bazı Metrikler

Karışıklık matrisinden hesaplanabilen bazı metrikler aşağıdaki gibi tanımlanabilir:

- **Doğruluk (Accuracy):**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Hassasiyet (Precision):**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Duyarlılık (Recall):**

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Skoru:**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

E. Sonuç

Karışıklık matrisi, modelin her sınıf için ne kadar doğru veya yanlış tahminler yaptığını anlamamıza yardımcı olur ve modelin güçlü ve zayıf yönlerini değerlendirmemize olanak tanır. Bu bilgiler, modelin iyileştirilmesi veya farklı hiperparametrelerle tekrar eğitilmesi için faydalı olabilir. Bizim modelimiz için Eğitim, Doğrulama ve Test veri setlerinin *Karışıklık matrisi*'ni Şekil 4'deki gibi elde ediyoruz

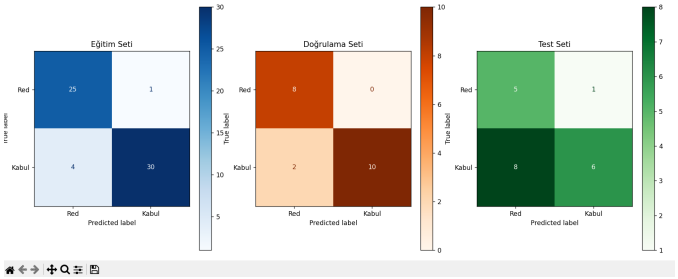

Fig. 4. Karışıklık Matrisleri

TABLE I
MODEL PERFORMANS METRIKLERI

Veri Seti	**Doğruluk**	**Kesinlik**	**Recall**	**F-Skor**
Eğitim	0.9167	0.9678	0.8824	0.9231
Doğrulama	0.90	1.0	0.8334	0.9090
Test	0.55	0.8575	0.4288	0.5715

V. SONUÇ

Bu çalışmada, lojistik regresyon yöntemi kullanılarak sınav sonuçlarına dayalı bir işe alım tahmin modeli geliştirilmiştir. Stochastic Gradient Descent algoritması ile eğitilen model, eğitim ve doğrulama setlerinde yüksek doğruluk ve F-skor değerleri göstermiştir. Fakat, test setinde olması gerkenden düşük sonuçlar göstermiştir. Ancak veri seti büyütülürse daha iyi sonuçlar elde edilebilir. Sonuçlar, yöntemin bu tür problemler için uygun olduğunu ve uygulamada kullanılabileceğini göstermektedir.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[2] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[3] F. Pedregosa, et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[4] T. Zhang, "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms," *Proceedings of the Twenty-first International Conference on Machine Learning (ICML-04)*, 2004.