

## 2025-2026 Güz Yarıyılı Makine Öğrenmesi Dersi 1. Ödevi

**Konu:** Lojistik Regresyon Yöntemi ile İkili Sınıflandırma

**Problem:** Bir firmaya iş başvurusunda bulunan kişilere yapılan mülakat ve iki sınav sonucuna göre iş alımları yapılmaktadır. Ekte verilen veri kümesinin her satırındaki ilk iki değer giriş bilgisi, üçüncü değer ise çıkış bilgisidir. Her satırda ilk iki değer farklı zamanlarda iş başvurusu yapan kişilerin iki sınav sonucunu, üçüncü değer ise işe kabul edilip edilmediklerini(kabul:1, ret:0) göstermektedir. Buna göre sınav notları verilen bir kişinin işe kabul edilip edilmeyeceğini lojistik regresyon yöntemi ile bulan (çıkış değeri için olasılık 0.50'den fazla ise kabul değilse ret) algoritmayı tasarlayınız ve kodlayınız.

### İşlem Adımları:

1. Ekte verilen örneklerin ilk %60'ını eğitim, sonraki %20'sini doğrulama, kalan %20'sini test için kullanınız.
2. Önce örneklerin iki sınıfa dağılımını görmek için x eksenini 1. sınav notu, y eksenini 2. sınav notu için kullanarak ve iki sınıfa ait örnekleri iki farklı renk ile göstererek **örnekleri** çizdiriniz.
3. Aşağıdaki bilgileri kullanarak lojistik regresyon işlemi yapan ve eğitim, doğrulama ve test örnekleri için *accuracy*, *precision*, *recall* ve *f-score* hesabını yapan **programı** Python'da yazınız. Program yazarken hazır fonksiyonları **kullanmayınız**.
  - a. **Aktivasyon fonksiyonu olarak sigmoid fonksiyonunu yazınız ve kullanınız.**
  - b. Ağırlık değerlerinin güncellenmesi için *stochastic gradient descent* kullanarak her örneğin hesapladığınız  $y_{predicted}$  çıkışı için ağırlıkları güncelleyiniz.
  - c. Loss fonksiyonu olarak **cross entropy loss** kullanınız.
    - i.  $CrossEntropyLoss = -[y_{target} * \log(y_{predicted}) + (1 - y_{target}) * \log(1 - y_{predicted})]$
    - ii. Her örnek için ortalama CrossEntropyLoss değerini hesaplayarak x eksenini örnek, y eksenini crossEntropyLoss'u gösterecek şekilde değişimi gösteren grafiği **çizdiriniz**. Bu grafiği inceleyerek öğrenmeyi ne zaman durdurmanız gerektiğini belirleyiniz.
  - d. Doğrulama örnekleri için CrossEntropyLoss'u hesaplayınız. Eğitim örnekleri için çizdirdiğiniz grafik üzerinde doğrulama örnekleri için de grafiği çizdirerek eğitim ve doğrulama örneklerinin birbirine göre durumunu değerlendirerek aşırı öğrenme olup olmadığını değerlendiriniz. Aşırı öğrenme varsa iyileştirme yapınız. Eğer iyileştirme ihtiyacı varsa nasıl yaptığınızı raporunuzda anlatınız ve yeni bir grafik üzerinde son durumu gösteriniz.
  - e. Eğitim, doğrulama ve test örnekleri için *accuracy*, *precision*, *recall* ve *f-score* hesabını yapınız.

## Değerlendirme:

Ödevler %60 kod ve %40 rapor olmak üzere 100 puan üzerinden değerlendirilecektir.

### **Kod (60 Puan)**

Bu ödevde lojistik regresyonda öğrenme için gerekli olan sigmoid, crossEntropyLoss ve ağırlık güncelleme işlemleri için gerekli kodu hazır fonksiyon kullanmadan sizin yazmanız istenmektedir. Fakat okuma/yazma, grafik çizdirme vb. işlemler için hazır kütüphanelerden faydalanabilirsiniz.

### **Kullanılan Kütüphaneler ve Gerekli Dosyalar**

Kodlar TensorFlow, Keras, Scikit-learn kullanılarak Python ortamında geliştirilmelidir.

ReadMe.txt dosyası içerisinde kodların nasıl çalıştırılabileceği kısaca açıklanmalıdır.

Örnek ReadMe.txt:

```
## Açıklama
Makine Öğrenmesi (BLM5110) dersi kapsamında TensorFlow ve Keras kullanılarak geliştirilen
sınıflandırma modeli.

## Gereksinimler
pip install -r requirements.txt

## Çalıştırma
Eğitim:
python train.py
Değerlendirme:
python eval.py

### Dosya Düzeni
proje_klasoru/
|- train.py
|- eval.py
|- requirements.txt
|- dataset/
|- results/
```

Gerekli kütüphaneler ve versiyonları requirements.txt dosyasında belirtilmelidir (Anaconda command prompt ile environment içerisindeyken 'conda list' komutu kullanılarak elde edilebilir).

ReadMe.txt ve requirements.txt dosyası eksikliğinde her birinden 10 puan kırılacaktır.

### **Kodun Modülerliği (20 Puan)**

- Kodlar sadece tek bir main fonksiyonundan oluşmamalı, modüler bir yapı izlenmelidir.
- Gerektiğinde kod, çeşitli scriptlere bölünmelidir (train.py, dataset.py, eval.py gibi).
- Kodun okunabilirliği ve anlaşılabilirliği açısından fonksiyon ve sınıflar kullanılmalıdır.
- Fonksiyon ve sınıflar, anlamlı isimlendirmeler ile tanımlanmalı ve kodun anlaşılabilirliği sağlanmalıdır.
- Fonksiyonlar docstring veya yorum satırları kullanılarak işlevleri ve parametreleri açıklanmalı.

Bir fonksiyon için örnek bir docstring:

```
def complex(real=0.0, imag=0.0):  
    """Bir karmaşık sayı oluştur.  
    Argümanlar:  
    real -- gerçek kısım (default 0.0)  
    imag -- sanal kısım (default 0.0)  
    """  
    if imag == 0.0 and real == 0.0:  
        return complex_zero
```

### Kodun Çalışabilirliği ve Testler (40 Puan)

- Kodun hatasız şekilde ilgili ödevde istenen çıktıları sağladığı ve çalışabilir olduğu test edilmelidir.
- Ara sonuçlar (epoch eğitim/validasyon loss bilgileri vb.) ve sonuçlar konsol üzerinde veya .txt formatında kaydedilerek sunulmalıdır.
- Çalışmadan alınan sonuçların yeterli bir başarı elde etmesi gerekmektedir.

### Rapor (40 Puan)

- Rapor, IEEE makale formatında hazırlanmalıdır ve 4 sayfayı geçmemelidir.
- Rapor akıcı olmalı, gereksiz açıklamalar yapılmamalı, görseller ve tablolar kullanılmalı ve yapılan değerlendirmeler olabildiğince kaynaklarla desteklenmelidir.
- Metin içerisinde görsellere ve tablolara kesinlikle referans verilmelidir.
- Yazım hataları, zaman kullanımı ve dil bilgisi kontrol edilmelidir.
- Rapor içeriği aşağıdaki ana bölümleri içermelidir:
  - Özet: Yaptığınız çalışmayı, elde ettiğiniz sonuçları bir paragraflık özet olarak veriniz.
  - Giriş: Ödev konusunu tanıtan bir paragraflık bir giriş yapınız. Problemin tanımı, çalışmanın amacı, çalışmanın kapsamı hakkında bilgi veriniz.
  - Deneysel Analiz: Yukarıda işlemler bölümünde belirtilen işlemler için elde ettiğiniz sonuçları tablo ve istenilen grafikleri kullanarak gösteriniz. Başarı değerlerini eğitim, doğrulama ve test için ayrı ayrı veriniz.
  - Sonuç: Elde ettiğiniz sonuçları bir paragrafta yorumlayınız.

**Önemli Not:** Yazdığınız kod ve raporunuz bir başka öğrencininkine veya internetteki kaynaklara belli bir orandan fazla benzerse kopya olarak değerlendirilecektir.

**Ödev Teslim Tarihi:** Ödevinizi **24 Kasım Pazartesi 23.00'a** kadar sisteme yükleyiniz. Lütfen sistemde olabilecek aksaklıkları da hesaba katarak ödevinizi sisteme yükleme işlemi son dakikalara bırakmayınız. Sistemde mücbir sebeplerden ötürü henüz kayıtlı olmayan öğrenciler dışında e-mail ile gönderilen ödevler kabul edilmeyecektir.