**ADEM MAVANACI**

**152120191040**

**LSTM MODEL (SPAM E POSTA SINIFLANDIRMASI - HW5) Raporu**

**TASK 1: Veri Seti Hazırlığı ve Yükleme**

**Yaptığım İşlemler:**

* emails.xlsx” dosyasından veri setini yükledim. Veri seti, “text” ve “spam” sütunlarından oluşuyordu.
* “text” sütununu e-postaların içerik bilgisi, “spam” sütununu ise etiket bilgisi (1: Spam, 0: Not Spam) olarak kullandım.
* E-postaları vektörlere dönüştürmek için CountVectorizer ile metin özelliklerini sayısallaştırdım. Özellik sayısını 5000 ile sınırladım.
* Veriyi %80 eğitim ve %20 test olarak böldüm.  
  **Karşılaştığım Sorunlar ve Çözümler:**

1. **Eksik Kolonlar:** Veri setinde “text” ve “spam” sütunlarının olmaması durumunda kullanıcıya hata mesajı göstermek için kontrol mekanizması ekledim.
2. **Dengesiz Veri:** Eğitim ve test verilerinde spam ve spam olmayan oranlarını eşitlemeye özen gösterdim.

**TASK 2: LSTM Modelinin Tasarımı**

**Yaptığım İşlemler:**

* PyTorch kullanarak LSTM tabanlı bir sınıflandırıcı modeli oluşturdum. Model:
  + Giriş boyutuna göre bir LSTM katmanı,
  + 128 gizli birim,
  + 2 tam bağlı (fully connected) çıkış biriminden oluşuyor.
* Modelin giriş boyutunu kontrol ederek, veriyi gerektiğinde 3 boyutlu hale getirdim.

**Karşılaştığım Sorunlar ve Çözümler:**

* **Giriş Boyutu Hatası:** Giriş verisinin boyutunun LSTM’nin beklediği formatta olmadığını fark ettim. Bu sorunu, giriş verisini unsqueeze yöntemiyle düzenleyerek çözdüm.

**TASK 3: Modelin Eğitimi**

**Yaptığım İşlemler:**

* Modeli eğitmek için aşağıdaki hiperparametreleri kullandım:
  + **Öğrenme Oranı (Learning Rate): 0.001**
  + **Batch Boyutu: 32**
  + **Epoch Sayısı: 10**
  + **Optimizasyon Algoritması: Adam**
* Eğitim sırasında kayıp değerlerini her epoch sonunda hesapladım.

Eğitim ve doğrulama setlerini kullanarak 10 epoch boyunca modeli eğittim. Eğitim sürecinde Train Loss ve Val Loss değerlerini takip ederek modelin performansını gözlemledim.

**Karşılaştığım Sorunlar ve Çözümler:**

**Eğitim Süresinin Uzunluğu:** Eğitim süresini kısaltmak için GPU kullanarak hesaplama hızını artırdım.

**TASK 4: Model Değerlendirmesi**

**Sonuçlar:**

Modelin test seti üzerindeki sonuçları şu şekildeydi:

**Confusion Matrix:**

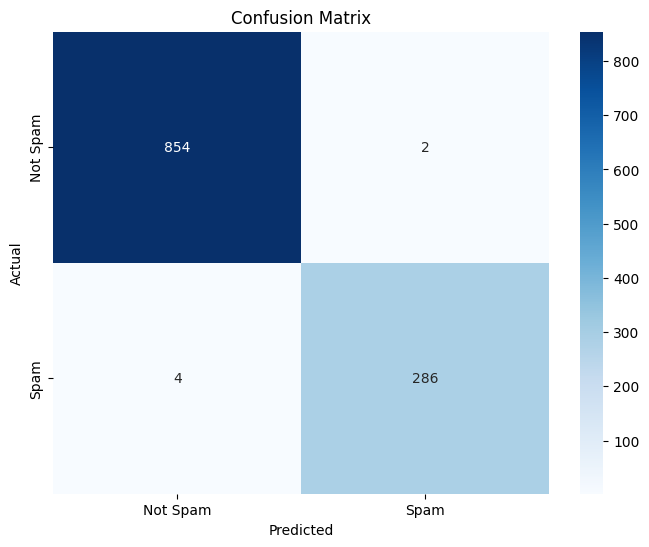
[[854 2]

[ 4 286]]

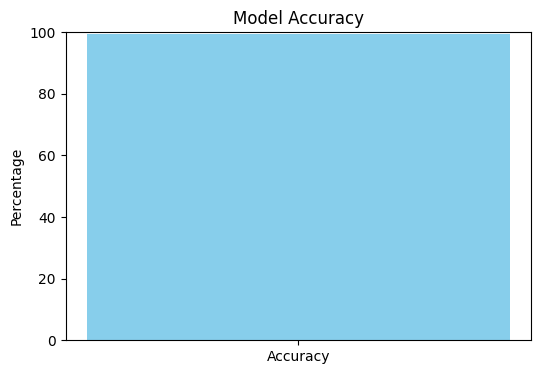
**Diğer Metrikler:**

* **Accuracy:** 99.48%
* **F1-Score:** 0.9948

**Confusion Matrix Heatmap:**

****

**Accuracy:**

****

**TASK 5: Modeli Kaydetme ve Yükleme**

**Yaptığım İşlemler:**

* Modeli torch.save fonksiyonuyla “lstm\_model.pth” dosyasına kaydettim.
* Kaydedilen modeli torch.load fonksiyonuyla yükledim ve test verilerinde değerlendirdim.

**Sonuç ve Yorumlar**

Proje kapsamında bir LSTM modeli geliştirdim ve spam e-posta sınıflandırması gerçekleştirdim. Model, test verisinde %99.48 doğruluk ve 0.9948 F1-Score ile başarılı bir sonuç verdi. İlerleyen çalışmalarda, modelin performansını artırmak için:

* Daha fazla veri eklenebilir,
* Hiperparametre optimizasyonu yapılabilir,
* Özellik sayısını artırmak için farklı vektörleştirme yöntemleri denenebilir.

**Ekran Görüntüleri:**

**metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu**