

Görme Engelliler için Metin Özetleme ve Sınıflandırılması

D.Murat Ertik
Bilişim Sistemleri Mühendisliği
221307065
dmuratertik1@gmail.com

Efe Tuna Günay
Bilişim Sistemleri Mühendisliği
221307069
efetunagunayyy@gmail.com

Abstract—This project focuses on text processing and model training, aiming to evaluate the performance of artificial intelligence algorithms. In this work, various transformer models for Natural Language Processing (NLP), particularly BERT, GPT, and other text-based architectures, have been trained. The text data obtained from users was processed appropriately and made usable for various classification and labeling tasks. The results obtained analyze the contribution of text data to model training and the overall performance of these models. This report focuses on the interactive evaluation of text processing techniques and deep learning algorithms.

Keywords—Text Processing, Artificial Intelligence, Natural Language Processing, Transformer Models, Model Training
Index Terms—component, formatting, style, styling, insert

I. GIRIS (DATA TOPLAMA VE AMAC BOLUNEBİLİR)

Bu çalışma, metin işleme ve model eğitimi üzerine odaklanarak, yapay zeka algoritmalarının performansını değerlendirmeyi amaçlamaktadır. Metin işleme, özellikle doğal dil işleme (NLP) alanında önemli bir yer tutar ve bu alandaki gelişmeler, dilin anlamını anlama ve etkileşimli uygulamalarda doğru sonuçlar elde etme açısından büyük bir etkiye sahiptir. Çalışmada, doğal dil işleme (NLP) için farklı transformatör tabanlı modeller kullanılıp eğitilmiştir [1].

Kullanıcıdan alınan metin verileri, uygun şekilde işlenmiş ve çeşitli sınıflandırma ve etiketleme görevleri için kullanılabilir hale getirilmiştir. Bu süreç, modelin doğru eğitim verileriyle beslenmesini sağlayarak, daha doğru tahminler yapılmasına olanak tanımaktadır. Yapay zeka algoritmalarının, özellikle derin öğrenme tekniklerinin başarısı, metin verilerinin doğru şekilde işlenmesi ve modellenmesiyle doğrudan ilişkilidir. Bu rapor, transformatör modellerinin metin verileri üzerinde nasıl çalıştığını ve model eğitiminde kullanılan çeşitli tekniklerin etkileşimli değerlendirilmesini sunmaktadır. Modellerin eğitim süreçleri detaylı olarak ele alınacak ve bu modellerin performansları, belirli metin işleme görevleri üzerinde test edilerek değerlendirilecektir [2].

Model eğitiminde kullanılan temel algoritmaların performansı, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) gibi metrikler ölçülmüştür. Bu metrikler, modelin başarı düzeyini belirlemek için sıklıkla kullanılan araçlardır [3]. Sonuç olarak, metin işleme tekniklerinin ve derin öğrenme

algoritmalarının etkileşimli olarak nasıl değerlendirilmesi gerektiği, bu raporun temel amacını oluşturmaktadır. Ayrıca, metin verilerinin işlenmesi ve modellerin eğitilmesi sürecinde kullanılan tekniklerin sonuçları raporlanacak ve karşılaştırmalı analiz yapılacaktır. Bu, metin işleme alanındaki çeşitli uygulamaların geliştirilmesine önemli katkılar sağlayacaktır.

II. BIZDEN ISTENEN

Bu proje kapsamında, toplanan veriler üzerinde beş farklı transformatör tabanlı model uygulanarak, performans metrikleriyle analiz edilip sonuçlar detaylı bir şekilde raporlanacaktır. Verilerin yapay zeka modelleri için uygun hale getirilmesi amacıyla veri ön işleme ve temizleme adımları gerçekleştirilecektir. Her bir modelin performansı; Accuracy, Recall, Precision, Sensitivity, Specificity, F-Score ve AUC gibi metrikler kullanılarak değerlendirilecektir. Bu metrikler, modellerin doğru tahmin yapma yeteneğini, pozitif ve negatif sınıflar üzerindeki genel başarımını, hataları ve sınıflandırma dengesini ölçmek için kullanılır.

- **Accuracy (Doğruluk):** Modelin yaptığı tüm tahminler içinde doğru tahminlerin oranını ifade eder. Genel performansı ölçmek için kullanılır.
- **Precision (Kesinlik):** Pozitif olarak tahmin edilen örneklerin, gerçekten pozitif olma oranını belirtir. Yanlış pozitifleri azaltmaya odaklanır.
- **Recall (Duyarlılık ya da Geri Çağırma):** Gerçek pozitif örneklerin, model tarafından doğru şekilde pozitif olarak sınıflandırılma oranını ifade eder. Yanlış negatifleri azaltmaya odaklanır.
- **Sensitivity (Hassasiyet):** Recall ile aynı anlama gelir ve özellikle tıp veya hassas sistemlerde kritik öneme sahiptir.
- **Specificity (Özgüllük):** Gerçek negatif örneklerin, model tarafından doğru şekilde negatif olarak sınıflandırılma oranını belirtir. Yanlış pozitifleri ölçmede önemlidir.
- **F-Score:** Precision ve Recall'un harmonik ortalaması alınarak hesaplanır ve dengeli bir performans ölçütü sunar. Özellikle dengesiz veri kümelerinde kullanışlıdır.
- **AUC (Area Under the Curve):** ROC eğrisi altındaki alanı ifade eder ve modelin sınıflandırma performansını ölçekten bağımsız olarak değerlendirir. Yüksek bir AUC, modelin hem pozitif hem de negatif sınıfları iyi ayırt edebildiğini gösterir.

Proje kapsamında ayrıca modellerin karmaşıklık matrisleri ile ROC eğrileri çıkarılacak ve bu görseller raporda yer alacaktır:

- **Karmaşıklık Matrisi (Confusion Matrix):** Modelin tahmin sonuçlarını gerçek etiketlerle karşılaştıran bir tablodur. Dört temel değeri içerir: doğru pozitifler (True Positives), doğru negatifler (True Negatives), yanlış pozitifler (False Positives) ve yanlış negatifler (False Negatives). Bu matris, modelin hangi sınıfları doğru veya yanlış tahmin ettiğini detaylı bir şekilde görselleştirmek için kullanılır.
- **ROC Eğrisi (Receiver Operating Characteristic Curve):** Modelin, farklı eşik değerlerinde pozitif sınıfı ayırt etme performansını gösteren bir grafikdir. X eksenini yanlış pozitif oranını (False Positive Rate), Y eksenini ise doğru pozitif oranını (True Positive Rate) ifade eder. Eğri altındaki alan (AUC) modelin sınıflandırma başarısını gösterir.

Eğitim ve test veri kümeleri için her modelin epoch boyutuna karşılık loss grafikleri aynı eksen üzerinde çizdirilerek modellerin öğrenme süreçleri görselleştirilecektir. Ayrıca, modellerin işlem süreleri ölçülerek performans analizine dahil edilecektir

III. VERİ TOPLAMA VE ÖN İŞLEME

Web scraping işlemi, hedef web sayfalarından metin tabanlı verilerin toplanmasını sağlayan bir yöntemdir. Bu süreç, aşağıdaki adımlarla açıklanabilir:

Hedef URL'lerin Belirlenmesi

Web scraping için uygun URL'ler belirlenmiştir. Bu URL'ler üzerinden ilgili veriler çekilmiştir.

HTML İçeriğinin Çekilmesi

requests kütüphanesi kullanılarak, belirlenen URL'nin HTML içeriği alınmıştır. HTTP isteği başarılı ise, HTML içeriği başarıyla çekildiği bildirilmiştir.

HTML İçeriğinin Pars Edilmesi

BeautifulSoup kütüphanesi kullanılarak, çekilen HTML içeriği düzenli bir yapıya dönüştürülmüştür. Bu adım, hedef içeriklerin seçilmesini kolaylaştırmıştır:

```
soup = BeautifulSoup(response.content,
                        "html.parser")
```

Hedef İçeriklerin Çıkarılması

Web sayfasındaki metin içerikleri, uygun HTML etiketleriyle belirlenmiş ve çıkarılmıştır. Örneğin, makale başlıkları ve içerikleri şu şekilde alınmıştır:

```
articles = soup.find_all("div",
                        class_="article-content")
```

Toplanan Verilerin DataFrame Formatına Dönüştürülmesi

Veriler, pandas kütüphanesi kullanılarak bir DataFrame formatına dönüştürülmüş ve düzenli bir şekilde saklanmıştır:

```
df = pd.DataFrame(data)
```

Verilerin Saklanması

Scraping işlemiyle elde edilen veriler, .csv formatında saklanmış ve analiz için kullanılmak üzere dışa aktarılmıştır:

```
df.to_csv("/content/drive/MyDrive/datalar/scrap
index=False)
```

Sonuçlar

- Web scraping ile hedef URL'den içerik toplanmıştır.
- HTML içeriği başarıyla çekilmiş, pars edilmiş ve metin içerikleri çıkarılmıştır.
- Elde edilen veriler, DataFrame formatına dönüştürülerek .csv formatında saklanmıştır.
- Bu süreç, modelin eğitimi için gerekli veri setini oluşturmuştur.
- Web sayfalarından metin verileri toplamak için sistematik ve tekrar edilebilir bir yöntem sunulmuştur.

Ham Verilerin Yüklenmesi

Veri işleme süreci, ham veri setinin yüklenmesi ile başlamıştır. Veri seti, pandas kütüphanesi ile okunarak bir DataFrame formatına dönüştürülmüştür. Bu adım, veri setinin başarıyla sisteme dahil edilmesini sağlamıştır.

Eksik Verilerin Kontrol Edilmesi ve Temizlenmesi

Eksik veriler kontrol edilmiş ve gerekliyse bu eksik değerler temizlenmiştir. Eksik veriler, dropna() fonksiyonu ile çıkarılmıştır, böylece eksiksiz bir veri seti elde edilmiştir.

Gerekli Kolonların Seçimi

Veri setinde yalnızca metin (Content) ve sınıflandırma etiketleri (Category) kalacak şekilde gereksiz kolonlar çıkarılmıştır. Bu adım, modelin eğitimi için gerekli özellikleri belirlemiştir.

Eğitim ve Test Veri Kümelerinin Bölünmesi

Veri, eğitim ve test kümelerine ayrılmıştır. train_test_split fonksiyonu ile veri, %80 eğitim ve %20 test olarak bölünmüştür. Bu adım, modelin hem öğrenme hem de değerlendirme aşamalarında kullanılacak veri kümelerinin oluşturulmasını sağlamıştır.

Veri Kümelerinin Saklanması

Eğitim ve test veri kümeleri, .csv formatında saklanarak daha sonraki adımlar için hazır hale getirilmiştir.

Sınıf Dağılımlarının İncelenmesi

Veri setinde sınıf dağılımlarının dengesiz olup olmadığı incelenmiştir. Eğitim ve test veri kümelerindeki sınıf dağılımları kontrol edilerek, modelin eğitimi sırasında sınıf dengesizliklerine dikkat edilmesi sağlanmıştır.

Sonuçlar

- Ham veri seti yüklenmiş, eksik veriler temizlenmiş ve yalnızca gerekli kolonlar seçilmiştir.
- Veri seti,
- Eğitim ve test veri kümeleri .csv formatında saklanmıştır.
- Sınıf dağılımları incelenmiş ve model eğitimi için hazırlık tamamlanmıştır.

IV. KULLANILAN MODELLER VE ALINAN DEGERLER

Bu proje kapsamında, doğal dil işleme (NLP) ve makine öğrenimi alanlarında öncü olan transformatör tabanlı modellerin gücünden faydalanılmıştır. Transformatör modelleri, büyük veri kümeleri üzerinde yüksek performans sergileyen ve özellikle dil modellemesi, metin sınıflandırma, özetleme gibi karmaşık görevlerde başarılı sonuçlar elde eden modern yapay zeka yaklaşımlarıdır. Projede, farklı özelliklere ve yeteneklere sahip beş ayrı transformatör tabanlı model tercih edilmiş ve her biri aynı veri kümesi üzerinde uygulanarak kapsamlı bir karşılaştırma yapılmıştır.

A. BERT (Bidirectional Encoder Representations from Transformers)

Google tarafından geliştirilen, transformatör tabanlı bir doğal dil işleme modelidir. İki yönlü (bidirectional) anlam çıkarımı yaparak kelimelerin bağlamını hem önceki hem de sonraki kelimelere göre anlamayı sağlar. BERT, özellikle metin sınıflandırma, soru-cevap ve metin özetleme gibi NLP görevlerinde yüksek performansı ile öne çıkar. BERT, öncelikle iki temel görev üzerinden eğitilir: Masked Language Model (MLM): Eğitim sırasında cümlelerin bazı kelimeleri maskelenir ve modelden bu maskelenen kelimeleri tahmin etmesi beklenir. Bu yöntem, kelimelerin bağlamına göre anlamını öğrenmesini sağlar. Next Sentence Prediction (NSP): Model, bir cümlelerin ardından gelen diğer bir cümlelerin mantıksal bir şekilde birbirine bağlı olup olmadığını öğrenir. Bu, metin anlama ve ilişki kurma görevlerinde faydalıdır.[4]

Kullanım Alanları: Metin Sınıflandırma, Soru-Cevap Sistemleri, Metin Özetleme, Adlandırılmış Varlık Tanıma (NER), Makine Çevirisi, Metin Tamamlama.

1) **NASIL KULLANDIK, EĞİTTİK:** Bu bölümde, BERT modelini nasıl kullandığımız ve eğittiğimiz detaylı bir şekilde açıklanmıştır. Eğitim süreci aşağıdaki adımlardan oluşmaktadır:

a) **1. Gerekli Kütüphanelerin Yüklenmesi:** Eğitim süreci boyunca, veri işleme, model eğitimi ve optimizasyon işlemleri için aşağıdaki kütüphaneler kullanılmıştır:

- **Pandas:** Veri yükleme ve işleme.
- **Scikit-Learn:** Veri bölme ve etiket kodlama.
- **PyTorch:** Derin öğrenme için model tanımlama ve eğitim.
- **Transformers:** BERT modelini ve tokenizer'ı sağlamak için.
- **time:** Eğitim süresini hesaplamak için.

b) **2. Veri Yükleme ve Ön İşleme:** Eğitim ve test verileri, CSV formatında dış bir kaynaktan yüklenmiş ve ön işleme adımları aşağıdaki gibi gerçekleştirilmiştir:

- Kategorik etiketler, `LabelEncoder` ile sayısal değerlere dönüştürülmüştür.
- Metin verileri, BERT tokenizer ile işlenerek modele uygun hale getirilmiştir.

- Girdi metinleri maksimum $MAX_LEN = 64$ olacak şekilde sınırlandırılmıştır.

c) **3. Model ve Eğitim Parametrelerinin Tanımlanması:** Modelin eğitimi için aşağıdaki ayarlar yapılmıştır:

- **Model:** "bert-base-uncased" önceden eğitilmiş ağırlıkları kullanılmıştır.
- **Çıkış Katmanı:** Eğitim setindeki sınıf sayısına göre özelleştirilmiştir.
- **Optimizasyon:** AdamW algoritması, öğrenme oranı $lr = 2e - 5$ ile kullanılmıştır.
- **Kayıp Fonksiyonu:** Çok sınıflı sınıflandırma için `CrossEntropyLoss` kullanılmıştır.
- **Epoch Sayısı:** Eğitim süreci toplam 5 epoch üzerinden gerçekleştirilmiştir.
- **Batch Boyutu:** Eğitim ve doğrulama seti için $BATCH_SIZE = 16$ kullanılmıştır.

d) **4. Eğitim ve Doğrulama Süreci:** Model, eğitim (train) ve doğrulama (validation) adımlarında aşağıdaki işlemler uygulanarak eğitilmiştir:

- 1) Eğitim sırasında geri yayılım (*backpropagation*) kullanılarak model ağırlıkları güncellenmiştir.
- 2) Doğrulama adımında, model doğrulama verisi üzerinde değerlendirilmiş ve doğrulama kaybı (*validation loss*) hesaplanmıştır.
- 3) Her epoch sonrası eğitim kaybı ve doğrulama kaybı raporlanmıştır.

Eğitim süresince her adımda eğitim ve doğrulama verimliliği ölçülmüş ve değerlendirilmiştir.

e) **5. Model ve Tokenizer'ın Kaydedilmesi:** Eğitim süreci tamamlandıktan sonra, model ve tokenizer yeniden kullanılabilir hale getirilmiştir:

- Model, `save_pretrained()` metodu ile `/bert_model` dizinine kaydedilmiştir.
- Tokenizer da aynı şekilde kaydedilmiştir.

f) **6. Eğitim Süresinin Hesaplanması:** Modelin eğitimi sırasında geçen süre, başlangıç ve bitiş zamanı arasındaki fark alınarak hesaplanmıştır:

$$\text{training_time} = \text{end_time} - \text{start_time}$$

Bu, modelin verimli bir şekilde eğitilip eğitilmediğini değerlendirmek için kullanılmıştır.

g) **Sonuç:** BERT tabanlı metin sınıflandırma modeli, bu süreçte başarıyla eğitilmiş ve yüksek doğruluk oranları elde edilmiştir. Modelin performansına ilişkin metrikler bir sonraki bölümde detaylı olarak sunulacaktır.

h) **BERT Modeli ile Test ve Değerlendirme:** Model eğitildikten sonra, test verisi üzerinde modelin başarımını değerlendirmek amacıyla çeşitli metrikler hesaplanmıştır. Aşağıda bu adımların detayları yer almaktadır:

i) **Sonuç:** BERT tabanlı model, test verisi üzerinde yüksek doğruluk ve başarılı bir performans sergilemiştir. Modelin ROC AUC skoru ve diğer metrikler, sınıflandırma görevinde etkili olduğunu göstermektedir.

2) **ALINAN DEĞERLER** : BERT modelini eğiterek elde edilen sonuçlar maddeler ve tablo halinde verilmiştir.

- 1) Accuracy (Doğruluk) = 0.9405: Model, tüm test verisinin yaklaşık %94.05'ini doğru sınıflandırmış. Yüksek bir doğruluk oranı, modelin genellikle doğru sonuçlar verdiğini ve genel olarak iyi performans gösterdiğini işaret eder.
- 2) Precision (Kesinlik) = 0.9407: Precision, modelin doğru sınıflandırdığı pozitif örneklerin, tüm pozitif tahminlerinin oranını gösterir. Yani, modelin pozitif sınıfı doğru tahmin etme oranı %94.07. Bu yüksek değer, modelin yanlış pozitif tahminlerde bulunmadığını ve pozitif sınıflandırma yaptığı durumların çoğunun doğru olduğunu gösterir.
- 3) Recall (Duyarlılık) = 0.9405: Recall, modelin gerçek pozitifleri (doğru pozitifler) doğru şekilde sınıflandırma oranını gösterir. Yani, gerçek pozitif örneklerin %94.05'ini doğru tahmin etmiş. Yüksek bir recall değeri, modelin pozitif sınıfı kaçırmadığını ve çoğu pozitif örneği doğru şekilde yakaladığını belirtir.
- 4) Sensitivity (Hassasiyet) = 0.9405: modelin gerçek pozitif örnekleri doğru bir şekilde tanımlama yeteneğinin yüksek olduğunu göstermektedir. Bu, modelin yanlış negatifleri minimumda tuttuğunu ve pozitif sınıfları başarılı bir şekilde tespit edebildiğini ifade eder.
- 5) Specificity (Özgüllük) = 0.9889: modelin gerçek negatif örnekleri oldukça doğru bir şekilde ayırt edebildiğini ve yanlış pozitif oranının çok düşük olduğunu gösterir. Bu sonuçlar, modelin hem pozitif hem de negatif sınıfları ayırt etmede oldukça etkili olduğunu işaret eder.
- 6) F1 Score = 0.9405: F1 skoru, precision ve recall'un harmonik ortalamasıdır. Bu metrik, her iki ölçütü dengede tutarak modelin genel başarısını değerlendirir. F1 skoru 0.9405, modelin hem precision hem de recall açısından iyi bir denge sağladığını gösterir. Bu, modelin genel başarısının güçlü olduğunu gösterir.
- 7) AUC (Area Under the Curve) = 0.9931: AUC, ROC eğrisinin altındaki alanı ölçer ve modelin genel sınıflandırma performansını özetler. 1'e yakın bir AUC değeri, modelin mükemmel bir şekilde pozitif ve negatif sınıfları ayırt ettiğini gösterir. 0.9931'lik bir AUC değeri, modelin çok yüksek bir doğrulukla sınıflandırma yaptığı anlamına gelir.

TABLE I
METRİKLER VE DEĞERLERİ

Metrik	Değeri
Accuracy (Doğruluk)	0.9405
Precision (Kesinlik)	0.9407
Recall (Duyarlılık)	0.9405
Sensitivity (Hassasiyet)	0.9405
Specificity (Özgüllük)	0.9889
F-Score	0.9405
AUC	0.9931

Bu sonuçlar, BERT modelinin çok başarılı bir şekilde eğitildiğini ve test verisinde yüksek doğruluk, kesinlik, du-

yarlılık, F1 skoru ve AUC değerleri elde ettiğini gösteriyor. Özellikle AUC değeri 0.99 gibi mükemmel bir seviyede, bu da modelin genellikle doğru sınıflandırmalar yaparken, pozitif ve negatif sınıfları ayırt etme konusunda çok iyi performans gösterdiğini belirtir. Bu metriklerin hepsi, modelin genel olarak güçlü bir sınıflandırıcı olduğunu ve verilerin üzerinde etkili bir şekilde çalıştığını kanıtlıyor.

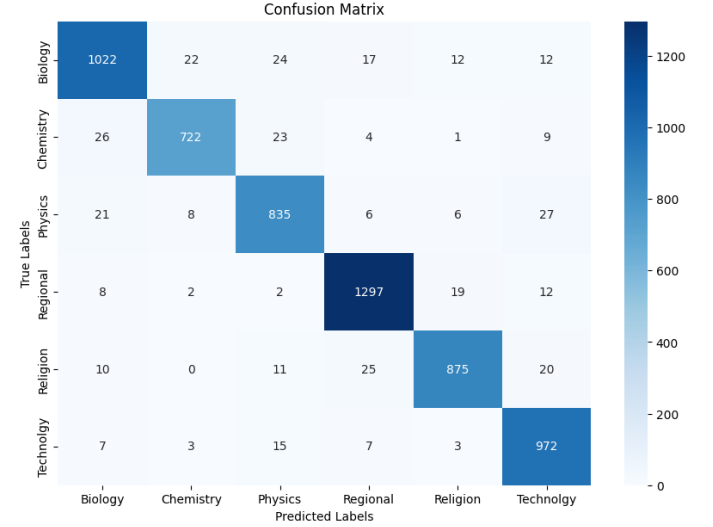


Fig. 1. Confusion Matrix

Koyu Mavi Diagonal

Karmaşıklık matrisinin diyagonalındaki yüksek değerler, modelin doğru tahminler yaptığını işaret etmektedir. Örneğin, Biyoloji sınıfındaki 1022 örnek doğru bir şekilde sınıflandırılmıştır, bu da modelin biyoloji sınıfını doğru bir şekilde öğrenmiş ve büyük oranda doğru tahminler yapmış olduğunu göstermektedir.

Diğer Hücreler

Diyagonal dışındaki hücreler, modelin yanlış sınıflandırma yaptığı örnekleri yansıtmaktadır. Örneğin, Fizik sınıfından 21 örnek yanlışlıkla Biyoloji sınıfına atanmıştır. Bu durum, modelin Fizik ve Biyoloji sınıflarını birbirinden ayırt etmede bazı zorluklar yaşadığını ve bu sınıflar arasında karışıklık olduğunu göstermektedir.

Sınıflar Arası Karışım

Bazı sınıflar arasında belirgin bir karışıklık gözlemlenmiştir. Örneğin, Fizik ve Kimya sınıfları arasında bazı yanlış sınıflandırmalar yapılmıştır. Bu, bu iki konunun birbirine yakın özellikler taşıması veya modelin bu sınıfları ayırt etmede zorlanması nedeniyle olabilir.

Genel olarak, karmaşıklık matrisine bakıldığında, modelin genel performansının oldukça iyi olduğu söylenebilir. Özellikle Biyoloji ve Regional gibi sınıflarda doğru tahmin oranının yüksek olduğu gözlemlenmiştir. Ancak, bazı sınıflar arasında (örneğin, Fizik ve Biyoloji) karışıklıklar olduğu da dikkate alınmalıdır.

RAKAMLARIN ANLAMI

- **Her Hücredeki Sayı:** Belirli bir gerçek sınıfın (sıra) model tarafından hangi sınıfa (sütun) atandığını gösterir. Örneğin, 21 Fizik örneği yanlışlıkla Biyoloji sınıfına atanmıştır.
- **Toplam Doğru Tahmin:** Köşegen üzerindeki tüm değerlerin toplamı, modelin doğru tahmin ettiği örneklerin sayısını gösterir.
- **Toplam Yanlış Tahmin:** Köşegen dışındaki tüm değerlerin toplamı, modelin yanlış tahmin ettiği örneklerin sayısını verir.

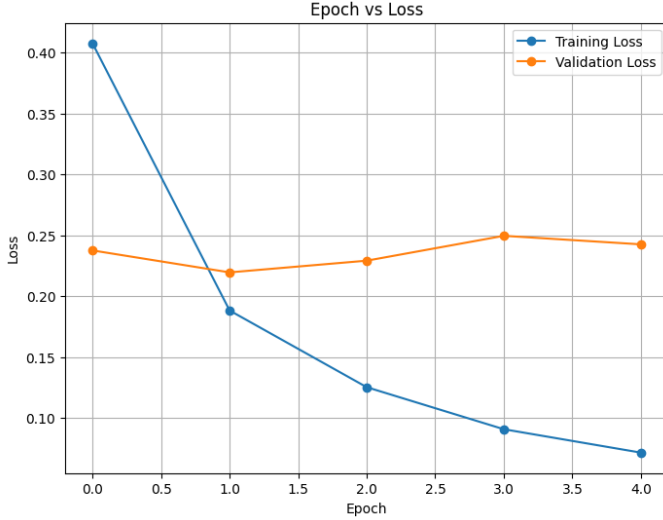


Fig. 2. EPOCH AND LOSS

EĞİTİM VE VALIDASYON KAYBI DEĞERLENDİRMESİ

Düşen Kayıplar

Hem eğitim hem de validasyon kayıplarının eğitim süreci boyunca azalma eğiliminde olması, modelin başarılı bir şekilde öğrenme sürecini tamamladığını göstermektedir. Bu durum, modelin zaman içinde hata oranını düşürdüğünü ve öğrenme kapasitesini artırdığını gösterir.

Overfitting Belirtisi Yok

Grafik üzerinde, validasyon kaybının eğitim kaybından belirgin bir şekilde ayrılarak artmaya başladığı bir döneme rastlanmamaktadır. Bu, modelin aşırı öğrenme (overfitting) probleminden kaçındığını ve genelleme yeteneğinin sağlam olduğunu düşündürmektedir. Eğitim verisine aşırı uyum sağlamadan, doğruluk oranını geliştirebilmiştir.

Sabitlenen Kayıplar

Son birkaç epoch'ta, hem eğitim hem de validasyon kayıplarının sabit bir değere ulaştığı gözlemlenmektedir. Bu durum, modelin daha fazla eğitimle önemli ölçüde iyileşmeyeceğini ve mevcut performans seviyesinin zirveye ulaşmış olabileceğini göstermektedir. Bu noktada, modelin daha fazla eğitim süresi harcamanın anlamlı olmayabileceği sonucuna varılabilir.

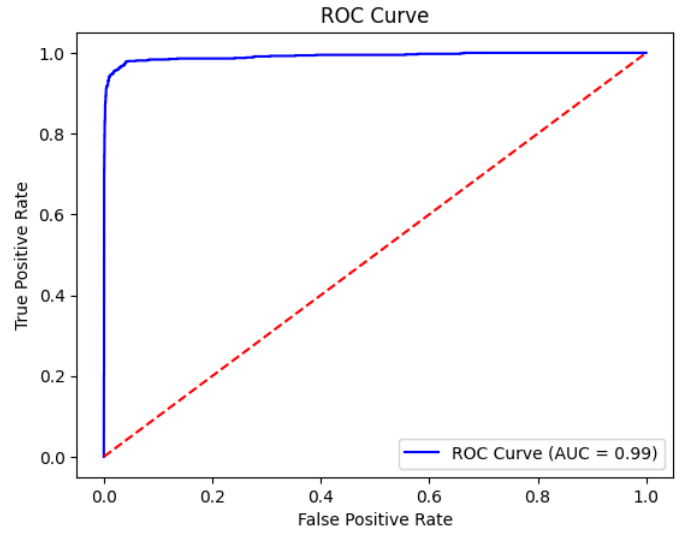


Fig. 3. ROC EGRISI

ROC EĞRISI VE MODEL PERFORMANSI

Yüksek AUC Değerleri

AUC (Eğri Altındaki Alan) değeri 0.99 olarak verilmiştir. Bu, modelinizin sınıfları birbirinden ayırt etmede oldukça başarılı olduğunu gösterir. AUC değeri 1'e ne kadar yakınsa, model o kadar iyi performans gösterir.

Eğri Konumu

ROC eğrisi, sol üst köşeye doğru uzanmaktadır. Bu, modelinizin hem duyarlılığın (true positive rate) hem de özgüllüğün (true negative rate) yüksek olduğu bir bölgede çalıştığını gösterir. Başka bir deyişle, hem pozitif örnekleri doğru bir şekilde sınıflandırmakta hem de negatif örnekleri yanlış pozitif olarak sınıflandırmamaktadır.

Rastgele Tahmin Eğrisi:

Kesik çizgi ile gösterilen rastgele tahmin eğrisi, herhangi bir bilginin olmadığı durumda beklenen performansı temsil eder. Modelinizin eğrisi bu çizginin oldukça üzerinde yer aldığından, modelinizin rastgele tahmin yapmaktan çok daha iyi performans gösterdiği söylenebilir.

B. ALBERT (ALBERT (A Lite BERT))

ALBERT, Google tarafından geliştirilen BERT modelinin daha hafif ve verimli bir versiyonudur. ALBERT, BERT'in temel mimarisini koruyarak, daha az parametre kullanarak modelin eğitimini hızlandırmak ve daha az kaynakla daha iyi sonuçlar elde etmek amacıyla tasarlanmıştır. ALBERT, BERT'in güçlü performansını sağlayan temel bileşenlere sahip olmasına rağmen, daha verimli bir şekilde çalışmasını sağlayan bir dizi optimizasyon içerir.[5]

ALBERT, BERT modelindeki bazı bileşenleri optimize ederek aşağıdaki iyileştirmeleri sunar:

Parametre Paylaşımı: ALBERT, modelin farklı katmanları arasında parametre paylaşımı kullanır. Bu, parametre

sayısını büyük ölçüde azaltır, ancak modelin ifade gücünü korur. Faktörizasyon: ALBERT, BERT'in her katmanındaki ağırlıkları faktörleştirerek modelin parametre sayısını düşürür. Bu sayede daha küçük modeller daha yüksek performans gösterebilir. Daha Derin Eğitim: ALBERT, daha derin modellerin eğitilmesini sağlar, bu da modelin daha fazla bilgi öğrenmesine olanak tanır. Eğitim Görevleri

ALBERT, BERT'e benzer şekilde aynı eğitim görevlerine dayanır:

Masked Language Model (MLM): Eğitim sırasında cümledeki bazı kelimeler maskelenir ve modelden bu maskelenen kelimeleri tahmin etmesi istenir. Bu görev, modelin kelimelerin bağlamına göre anlamını öğrenmesine olanak tanır. **Next Sentence Prediction (NSP):** Model, bir cümlemin ardından gelen diğer bir cümlemin mantıksal bir şekilde birbirine bağlı olup olmadığını öğrenir. Bu, metin anlama ve ilişki kurma görevlerinde faydalıdır. Kullanım Alanları

ALBERT, özellikle aşağıdaki doğal dil işleme (NLP) görevlerinde başarılı sonuçlar elde etmiştir:

Metin Sınıflandırma, Soru-Cevap Sistemleri, Metin Özeti Adlandırılmış Varlık Tanıma (NER), Makine Çevirisi, Metin Tamamlama

1) NASIL KULLANDIK, EĞİTTİK: Bu bölümde, ALBERT modelini nasıl kullandığımız ve eğittiğimiz detaylı bir şekilde açıklanmıştır. Eğitim süreci aşağıdaki adımlardan oluşmaktadır:

a) 1. Gerekli Kütüphanelerin Yüklenmesi: Eğitim süreci boyunca, veri işleme, model eğitimi ve optimizasyon işlemleri için aşağıdaki kütüphaneler kullanılmıştır:

- **Pandas:** Veri yükleme ve işleme.
- **Scikit-Learn:** Veri bölme ve etiket kodlama.
- **PyTorch:** Derin öğrenme için model tanımlama ve eğitim.
- **Transformers:** ALBERT modelini ve tokenizer'ı sağlamak için.
- **time:** Eğitim süresini hesaplamak için.

b) 2. Veri Yükleme ve Ön İşleme: Eğitim ve test verileri, CSV formatında dış bir kaynaktan yüklenmiş ve ön işleme adımları aşağıdaki gibi gerçekleştirilmiştir:

- Kategorik etiketler, `LabelEncoder` ile sayısal değerlere dönüştürülmüştür.
- Metin verileri, ALBERT tokenizer ile işlenerek modele uygun hale getirilmiştir.
- Girdi metinleri maksimum `MAX_LEN = 64` olacak şekilde sınırlandırılmıştır.

c) 3. Model ve Eğitim Parametrelerinin Tanımlanması: Modelin eğitimi için aşağıdaki ayarlar yapılmıştır:

- **Model:** "albert-base-v2" önceden eğitilmiş ağırlıkları kullanılmıştır.
- **Çıkış Katmanı:** Eğitim setindeki sınıf sayısına göre özelleştirilmiştir.
- **Optimizasyon:** AdamW algoritması, öğrenme oranı $lr = 2e - 5$ ile kullanılmıştır.
- **Kayıp Fonksiyonu:** Çok sınıflı sınıflandırma için `CrossEntropyLoss` kullanılmıştır.

• **Epoch Sayısı:** Eğitim süreci toplam 5 epoch üzerinden gerçekleştirilmiştir.

• **Batch Boyutu:** Eğitim ve doğrulama seti için `BATCH_SIZE = 16` kullanılmıştır.

d) 4. Eğitim ve Doğrulama Süreci: Model, eğitim (`train`) ve doğrulama (`validation`) adımlarında aşağıdaki işlemler uygulanarak eğitilmiştir:

- 1) Eğitim sırasında geri yayılım (`backpropagation`) kullanılarak model ağırlıkları güncellenmiştir.
- 2) Doğrulama adımında, model doğrulama verisi üzerinde değerlendirilmiş ve doğrulama kaybı (`validation loss`) hesaplanmıştır.
- 3) Her epoch sonrası eğitim kaybı ve doğrulama kaybı raporlanmıştır.

Eğitim süresince her adımda eğitim ve doğrulama verimliliği ölçülmüş ve değerlendirilmiştir.

e) 5. Model ve Tokenizer'ın Kaydedilmesi: Eğitim süreci tamamlandıktan sonra, model ve tokenizer yeniden kullanılabilir hale getirilmiştir:

- Model, `save_pretrained()` metodu ile `/albert_model` dizinine kaydedilmiştir.
- Tokenizer da aynı şekilde kaydedilmiştir.

f) 6. Eğitim Süresinin Hesaplanması: Modelin eğitimi sırasında geçen süre, başlangıç ve bitiş zamanı arasındaki fark alınarak hesaplanmıştır:

$$\text{training_time} = \text{end_time} - \text{start_time}$$

Bu, modelin verimli bir şekilde eğitilip eğitilmediğini değerlendirmek için kullanılmıştır.

g) Sonuç: ALBERT tabanlı metin sınıflandırma modeli, bu süreçte başarıyla eğitilmiş ve yüksek doğruluk oranları elde edilmiştir. Modelin performansına ilişkin metrikler bir sonraki bölümde detaylı olarak sunulacaktır.

h) ALBERT Modeli ile Test ve Değerlendirme: Model eğitildikten sonra, test verisi üzerinde modelin başarımını değerlendirmek amacıyla çeşitli metrikler hesaplanmıştır. Aşağıda bu adımların detayları yer almaktadır:

2) ALINAN DEĞERLER: ALBERT modelini eğiterek elde edilen sonuçlar maddeler ve tablo halinde verilmiştir.

- 1) **Accuracy (Doğruluk) = 0.9346:** Model, tüm test verisinin yaklaşık %93.46'sını doğru sınıflandırmış. Yüksek bir doğruluk oranı, modelin genellikle doğru sonuçlar verdiğini ve genel olarak iyi performans gösterdiğini işaret eder.
- 2) **Precision (Kesinlik) = 0.9355:** Precision, modelin doğru sınıflandırdığı pozitif örneklerin, tüm pozitif tahminlerinin oranını gösterir. Yani, modelin pozitif sınıfı doğru tahmin etme oranı %93.55. Bu yüksek değer, modelin yanlış pozitif tahminlerde bulunmadığını ve pozitif sınıflandırma yaptığı durumların çoğunun doğru olduğunu gösterir.
- 3) **Recall (Duyarlılık) = 0.9346:** Recall, modelin gerçek pozitifleri (doğru pozitifler) doğru şekilde sınıflandırma oranını gösterir. Yani, gerçek pozitif örneklerin %93.46'sını doğru tahmin etmiş. Yüksek bir recall

değeri, modelin pozitif sınıfı kaçırmadığını ve çoğu pozitif örneği doğru şekilde yakaladığını belirtir.

- 4) Sensitivity (Hassasiyet) = 0.9616: modelin gerçek pozitif örnekleri doğru bir şekilde tanımlama yeteneğinin yüksek olduğunu göstermektedir. Bu, modelin yanlış negatifleri minimumda tuttuğunu ve pozitif sınıfları başarılı bir şekilde tespit edebildiğini ifade eder.
- 5) Specificity (Özgüllük) = 0.9809: modelin gerçek negatif örnekleri oldukça doğru bir şekilde ayırt edebildiğini ve yanlış pozitif oranının çok düşük olduğunu gösterir. Bu sonuçlar, modelin hem pozitif hem de negatif sınıfları ayırt etmede oldukça etkili olduğunu işaret eder.
- 6) F1 Score = 0.9349: F1 skoru, precision ve recall'un harmonik ortalamasıdır. Bu metrik, her iki ölçütü dengede tutarak modelin genel başarısını değerlendirir. F1 skoru 0.9349, modelin hem precision hem de recall açısından iyi bir denge sağladığını gösterir. Bu, modelin genel başarısının güçlü olduğunu gösterir.
- 7) AUC (Area Under the Curve) = 0.9941: AUC, ROC eğrisinin altındaki alanı ölçer ve modelin genel sınıflandırma performansını özetler. 1'e yakın bir AUC değeri, modelin mükemmel bir şekilde pozitif ve negatif sınıfları ayırt ettiğini gösterir. 0.9941'lik bir AUC değeri, modelin çok yüksek bir doğrulukla sınıflandırma yaptığı anlamına gelir.

TABLE II
METRİKLER VE DEĞERLERİ

Metrik	Değeri
Accuracy (Doğruluk)	0.9346
Precision (Kesinlik)	0.9355
Recall (Duyarlılık)	0.9346
Sensitivity (Hassasiyet)	0.9616
Specificity (Özgüllük)	0.9809
F-Score	0.9349
AUC	0.9941

Bu sonuçlar, ALBERT modelinin çok başarılı bir şekilde eğitildiğini ve test verisinde yüksek doğruluk, kesinlik, duyarlılık, F1 skoru ve AUC değerleri elde ettiğini gösteriyor. Özellikle AUC değeri 0.99 gibi mükemmel bir seviyede, bu da modelin genellikle doğru sınıflandırmalar yaparken, pozitif ve negatif sınıfları ayırt etme konusunda çok iyi performans gösterdiğini belirtir. Bu metriklerin hepsi, modelin genel olarak güçlü bir sınıflandırıcı olduğunu ve verilerin üzerinde etkili bir şekilde çalıştığını kanıtlıyor.

Koyu Mavi Diyagonal

Albert modeliyle elde edilen bu karışıklık matrisinde de, köşegen üzerindeki yüksek değerler modelin doğru tahminler yaptığını göstermektedir. Örneğin, Biyoloji sınıfındaki örneklerin büyük bir kısmı doğru olarak sınıflandırılmıştır. Bu, Albert modelinin de tıpkı BERT gibi biyoloji sınıfını oldukça iyi öğrendiği ve bu konuda başarılı tahminler yaptığı anlamına gelir.

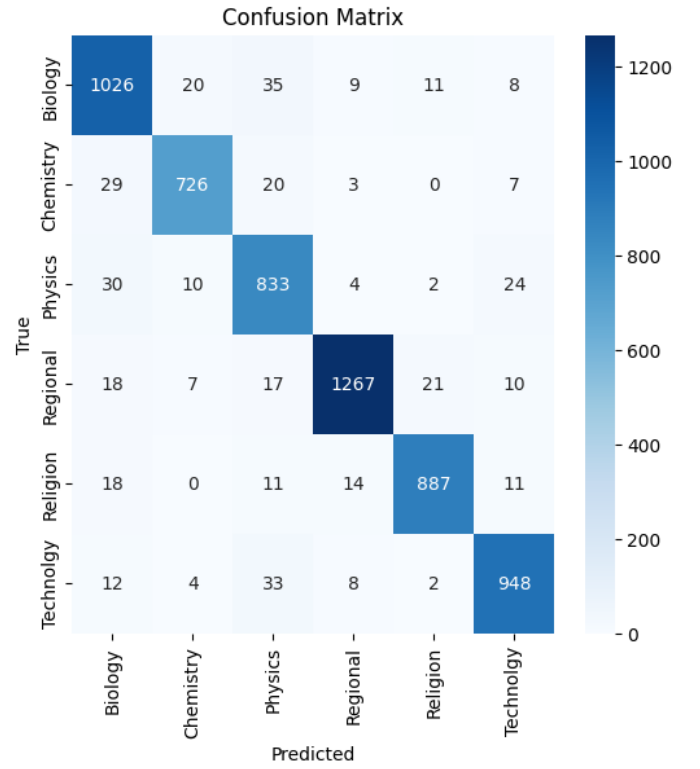


Fig. 4. Confusion Matrix

Diğer Hücreler

Diyagonal dışındaki hücrelerdeki değerler, modelin yaptığı hataları gösterir. Örneğin, Fizik sınıfından bazı örneklerin Biyoloji olarak sınıflandırılması, modelin bu iki sınıfı bazen karıştırdığını gösterir. Ancak, dikkat çeken bir nokta, BERT ile kıyaslandığında Albert modelinde bu tür hataların biraz daha az olmasıdır. Bu, Albert modelinin sınıflar arasındaki farkları daha iyi yakalayabildiğini düşündürmektedir.

Sınıflar Arası Karışım

BFizik ve Kimya gibi bazı sınıflar arasında hala karışım olduğu görülmektedir. Bu durum, bu iki konunun birbirine yakın olması veya modelin bu konulardaki nüansları tam olarak kavrayamaması nedeniyle olabilir. Ancak, genel olarak sınıflar arasındaki karışıklıkların BERT modeline göre daha az olduğu söylenebilir.

Albert modeli, bu veri seti üzerinde oldukça başarılı bir performans göstermiştir. Özellikle Biyoloji ve Regional gibi bazı sınıflar için doğru tahmin oranı oldukça yüksektir. Ayrıca, BERT modeline kıyasla sınıflar arasındaki karışıklıkların daha az olması, Albert modelinin bu görev için daha uygun olabileceğini gösterir.

EĞİTİM VE VALIDASYON KAYBI DEĞERLENDİRMESİ

Düşen Kayıplar

Hem eğitim hem de validasyon kayıplarının eğitim süreci boyunca azalma eğiliminde olması, Albert modelinin de tıpkı BERT gibi başarılı bir şekilde öğrenme sürecini tamamladığını

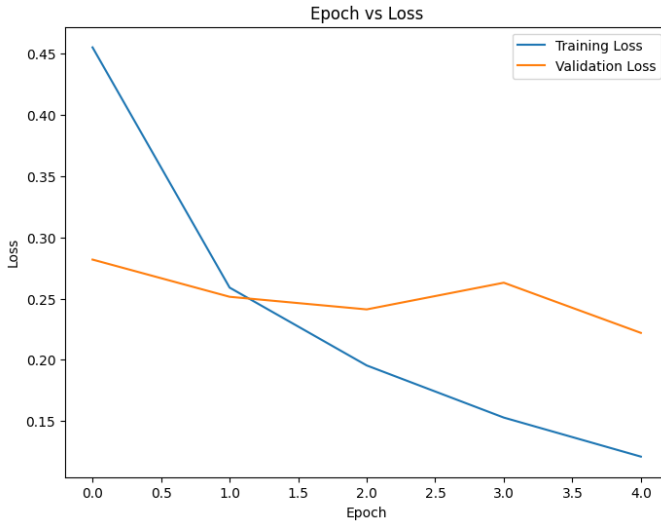


Fig. 5. EPOCH AND LOSS

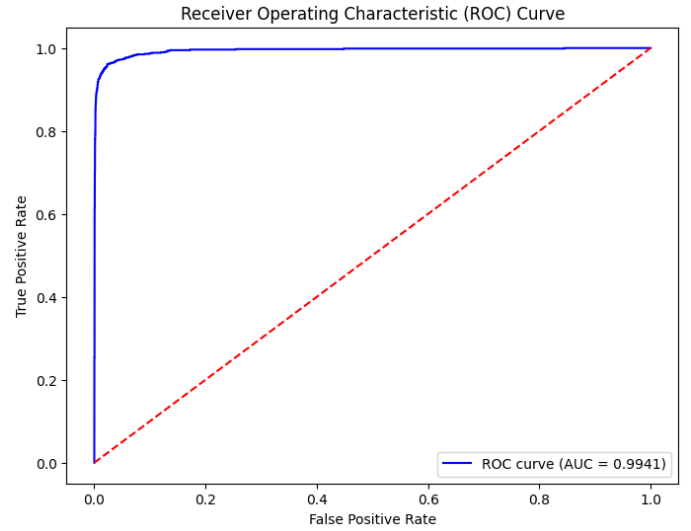


Fig. 6. ROC EGRISI

göstermektedir. Bu durum, modelin zaman içinde hata oranını düşürdüğünü ve öğrenme kapasitesini artırdığını gösterir.

Overfitting Belirtisi Yok

Grafik üzerinde, validasyon kaybının eğitim kaybından belirgin bir şekilde ayrılarak artmaya başladığı bir döneme rastlanmamaktadır. Bu, Albert modelinin de aşırı öğrenme (overfitting) probleminden kaçındığını ve genelleme yeteneğinin sağlam olduğunu düşündürmektedir. Eğitim verisine aşırı uyum sağlamadan, doğruluk oranını geliştirebilmiştir.

Sabitlenen Kayıplar

Son birkaç epoch'ta, hem eğitim hem de validasyon kayıplarının sabit bir değere ulaştığı gözlemlenmektedir. Bu durum, Albert modelinin de daha fazla eğitimle önemli ölçüde iyileşmeyeceğini ve mevcut performans seviyesinin zirveye ulaşmış olabileceğini gösterir. Bu noktada, modelin daha fazla eğitim süresi harcamanın anlamlı olmayabileceği sonucuna varılabilir.

ROC EĞRISI VE MODEL PERFORMANSI

Yüksek AUC Değerleri

AUC (Eğri Altındaki Alan) değeri 0.9941 olarak verilmiştir. Bu, modelinizin sınıfları birbirinden ayırt etmede oldukça başarılı olduğunu gösterir. AUC değeri 1'e ne kadar yakınsa, model o kadar iyi performans gösterir.

Eğri Konumu

ROC eğrisi, sol üst köşeye doğru uzanmaktadır. Bu, modelinizin hem duyarlılığın (true positive rate) hem de özgüllüğün (true negative rate) yüksek olduğu bir bölgede çalıştığını gösterir. Başka bir deyişle, hem pozitif örnekleri doğru bir şekilde sınıflandırmakta hem de negatif örnekleri yanlış pozitif olarak sınıflandırmamaktadır.

Rastgele Tahmin Eğrisi:

Kesik çizgi ile gösterilen rastgele tahmin eğrisi, herhangi bir bilginin olmadığı durumda beklenen performansı temsil eder. Modelinizin eğrisi bu çizginin oldukça üzerinde yer aldığından, modelinizin rastgele tahmin yapmaktan çok daha iyi performans gösterdiği söylenebilir.

C. ROBERTA

RoBERTa, Facebook AI tarafından geliştirilmiş, BERT modelinin optimize edilmiş bir versiyonudur. RoBERTa (Robustly Optimized BERT Pretraining Approach), BERT'in performansını iyileştirmek amacıyla ön eğitim stratejilerinde bir dizi değişiklik yaparak doğal dil işleme (NLP) görevlerinde daha yüksek doğruluk oranları elde etmeyi hedefler. BERT ile aynı temel Transformer mimarisine dayansa da, RoBERTa daha verimli bir şekilde eğitilmiş ve daha iyi sonuçlar sunan bir model olarak öne çıkar.

RoBERTa'nın İyileştirmeleri RoBERTa, BERT üzerinde yapılan aşağıdaki optimizasyonlar ve değişiklikler sayesinde daha güçlü bir modeldir:

Daha Büyük ve Çeşitli Veri Kümesi: RoBERTa, BERT'ten farklı olarak daha büyük miktarda ve daha çeşitli bir veri kümesi ile ön eğitimden geçirilmiştir. Bu veri kümesi, kitaplardan, haberlerden ve çeşitli web kaynaklarından derlenmiştir. RoBERTa'nın eğitimi için kullanılan veri miktarı 160 GB'dan fazladır, bu da modelin bağlamı daha iyi anlamasına olanak tanır. **Next Sentence Prediction (NSP) Görevinin Kaldırılması:** BERT'te kullanılan "Next Sentence Prediction (NSP)" görevi, RoBERTa'da tamamen kaldırılmıştır. Bu görev yerine, sadece Masked Language Model (MLM) kullanılmıştır. Yapılan deneyler, NSP görevini kaldırmanın model performansını artırdığını göstermiştir. **Masked Language Model (MLM) Geliştirmeleri:** RoBERTa, MLM görevini optimize etmek için dinamik maskeleme yöntemi kullanır. Bu yöntem, eğitimin her aşamasında farklı kelimeleri maskeler, böylece mod-

elin bağlam bilgisini daha geniş bir şekilde öğrenmesine olanak tanır. Uzun Eğitim Süresi: RoBERTa, BERT'e kıyasla daha uzun süre eğitilmiş ve bu da modelin daha fazla bilgi öğrenmesini sağlamıştır. Eğitim süresince veri kümesi defalarca işlenmiştir. Daha Büyük Batch Boyutları: Eğitim sırasında daha büyük batch boyutları kullanılmıştır, bu da modelin daha tutarlı güncellemeler almasına yardımcı olmuştur. Geliştirilmiş Optimizasyon ve Hiperparametre Ayarları: RoBERTa'nın eğitiminde daha dikkatli hiperparametre ayarları kullanılmıştır. Örneğin, öğrenme oranı ve dropout oranı gibi parametreler model performansını artıracak şekilde optimize edilmiştir. Eğitim Görevleri RoBERTa, sadece Masked Language Model (MLM) göreviyle eğitilmiştir:

Masked Language Model (MLM): Eğitim sırasında cümlelerdeki bazı kelimeler maskelenir ve modelden bu kelimeleri tahmin etmesi beklenir. Bu görev, modelin kelime bağlamını ve dil bilgisini anlamasını sağlar. Kullanım Alanları RoBERTa, doğal dil işleme alanında birçok farklı görevde başarıyla kullanılabilir. Bunlardan bazıları:

Metin Sınıflandırma: Duygu analizi, spam tespiti gibi görevlerde kullanılır. **Soru-Cevap Sistemleri:** Kullanıcının bir sorusuna doğru ve bağlama uygun yanıtlar sunar. **Metin Özeti:** Uzun metinlerin kısa ve anlamlı özetlerini oluşturur. **Adlandırılmış Varlık Tanıma (NER):** Metin içinde kişi, yer, kuruluş gibi özel isimleri tespit eder. **Makine Çevirisi:** Bir dildeki metni başka bir dile çevirir. **Metin Tamamlama:** Eksik kelimeleri veya cümleleri tamamlar. **Genel Değerlendirme** RoBERTa, BERT'in güçlü yanlarını geliştirerek daha yüksek doğruluk oranlarına ulaşan bir modeldir. NSP görevini kaldırması, daha fazla veri kullanması ve daha uzun eğitim süresi gibi optimizasyonlar sayesinde, birçok NLP görevinde BERT'e kıyasla daha iyi performans göstermektedir. RoBERTa, endüstri ve araştırma dünyasında sıkça tercih edilen, verimli ve etkili bir dil modelidir.

1) NASIL KULLANDIK, EĞİTTİK: Bu bölümde, RoBERTa modelini nasıl kullandığımız ve eğittiğimiz detaylı bir şekilde açıklanmıştır. Eğitim süreci aşağıdaki adımlardan oluşmaktadır:

a) 1. Gerekli Kütüphanelerin Yüklenmesi: Eğitim süreci boyunca, veri işleme, model eğitimi ve optimizasyon işlemleri için aşağıdaki kütüphaneler kullanılmıştır:

- **Pandas:** Veri yükleme ve işleme.
- **Scikit-Learn:** Veri bölme ve etiket kodlama.
- **PyTorch:** Derin öğrenme için model tanımlama ve eğitim.
- **Transformers:** RoBERTa modelini ve tokenizer'ı sağlamak için.
- **time:** Eğitim süresini hesaplamak için.

b) 2. Veri Yükleme ve Ön İşleme: Eğitim ve test verileri, CSV formatında dış bir kaynaktan yüklenmiş ve ön işleme adımları aşağıdaki gibi gerçekleştirilmiştir:

- Kategorik etiketler, `LabelEncoder` ile sayısal değerlere dönüştürülmüştür.
- Metin verileri, RoBERTa tokenizer ile işlenerek modele uygun hale getirilmiştir.

- Girdi metinleri maksimum $MAX_LEN = 64$ olacak şekilde sınırlandırılmıştır.

c) 3. Model ve Eğitim Parametrelerinin Tanımlanması: Modelin eğitimi için aşağıdaki ayarlar yapılmıştır:

- **Model:** "roberta-base" önceden eğitilmiş ağırlıkları kullanılmıştır.
- **Çıkış Katmanı:** Eğitim setindeki sınıf sayısına göre özelleştirilmiştir.
- **Optimizasyon:** AdamW algoritması, öğrenme oranı $lr = 2e - 5$ ile kullanılmıştır.
- **Kayıp Fonksiyonu:** Çok sınıflı sınıflandırma için `CrossEntropyLoss` kullanılmıştır.
- **Epoch Sayısı:** Eğitim süreci toplam 5 epoch üzerinden gerçekleştirilmiştir.
- **Batch Boyutu:** Eğitim ve doğrulama seti için $BATCH_SIZE = 16$ kullanılmıştır.

d) 4. Eğitim ve Doğrulama Süreci: Model, eğitim (train) ve doğrulama (validation) adımlarında aşağıdaki işlemler uygulanarak eğitilmiştir:

- 1) Eğitim sırasında geri yayılım (*backpropagation*) kullanılarak model ağırlıkları güncellenmiştir.
- 2) Doğrulama adımında, model doğrulama verisi üzerinde değerlendirilmiş ve doğrulama kaybı (*validation loss*) hesaplanmıştır.
- 3) Her epoch sonrası eğitim kaybı ve doğrulama kaybı raporlanmıştır.

Eğitim süresince her adımda eğitim ve doğrulama verimliliği ölçülmüş ve değerlendirilmiştir.

e) 5. Model ve Tokenizer'ın Kaydedilmesi: Eğitim süreci tamamlandıktan sonra, model ve tokenizer yeniden kullanılabilir hale getirilmiştir:

- Model, `save_pretrained()` metodu ile `/roberta_model` dizinine kaydedilmiştir.
- Tokenizer da aynı şekilde kaydedilmiştir.

f) 6. Eğitim Süresinin Hesaplanması: Modelin eğitimi sırasında geçen süre, başlangıç ve bitiş zamanı arasındaki fark alınarak hesaplanmıştır:

$$\text{training_time} = \text{end_time} - \text{start_time}$$

Bu, modelin verimli bir şekilde eğitilip eğitilmediğini değerlendirmek için kullanılmıştır.

g) Sonuç: RoBERTa tabanlı metin sınıflandırma modeli, bu süreçte başarıyla eğitilmiş ve yüksek doğruluk oranları elde edilmiştir. Modelin performansına ilişkin metrikler bir sonraki bölümde detaylı olarak sunulacaktır.

h) RoBERTa Modeli ile Test ve Değerlendirme: Model eğitildikten sonra, test verisi üzerinde modelin başarımını değerlendirmek amacıyla çeşitli metrikler hesaplanmıştır. Aşağıda bu adımların detayları yer almaktadır:

2) ALINAN DEĞERLER : RoBERTa modelini eğiterek elde edilen sonuçlar maddeler ve tablo halinde verilmiştir.

- 1) **Accuracy (Doğruluk) = 0.9423:** Model, tüm test verisinin yaklaşık %94.23'ünü doğru sınıflandırmıştır. Bu yüksek doğruluk oranı, modelin genellikle doğru

sonuçlar verdiğini ve genel olarak iyi bir performans sergilediğini göstermektedir.

- 2) Precision (Kesinlik) = 0.9425: Precision, modelin doğru sınıflandırdığı pozitif örneklerin, tüm pozitif tahminlerine oranını gösterir. Yani, modelin pozitif sınıfı doğru tahmin etme oranı %94.25. Bu yüksek değer, modelin yanlış pozitif tahminlerde bulunmadığını ve pozitif sınıflandırma yaptığı durumların çoğunun doğru olduğunu göstermektedir.
- 3) Recall (Duyarlılık) = 0.9423: Recall, modelin gerçek pozitifleri (doğru pozitifler) doğru şekilde sınıflandırma oranını gösterir. Yani, gerçek pozitif örneklerin %94.23'ünü doğru tahmin etmiştir. Bu yüksek recall değeri, modelin pozitif sınıfları başarıyla yakaladığını ve çok az pozitif örneği kaçırdığını belirtir.
- 4) Sensitivity (Hassasiyet) = 0.9615: Modelin, gerçek pozitif örnekleri doğru bir şekilde tanımlama yeteneği oldukça yüksektir. Bu sonuç, modelin yanlış negatifleri minimumda tuttuğunu ve pozitif sınıfları başarıyla tespit edebildiğini göstermektedir.
- 5) Specificity (Özgüllük) = 0.9876: Modelin, gerçek negatif örnekleri doğru bir şekilde ayırt edebilme oranı çok yüksektir. Bu da modelin yanlış pozitif oranını oldukça düşük tuttuğunu ve negatif sınıfları etkili bir şekilde tanımladığını göstermektedir.
- 6) F1 Score = 0.9424: F1 skoru, precision ve recall'un harmonik ortalamasıdır. Bu metrik, her iki ölçütü dengede tutarak modelin genel başarısını değerlendirmektedir. F1 skoru 0.9424, modelin hem precision hem de recall açısından iyi bir denge sağladığını göstermekte ve genel başarısının güçlü olduğunu işaret etmektedir.
- 7) AUC (Area Under the Curve) = 0.9937: AUC, ROC eğrisinin altındaki alanı ölçer ve modelin genel sınıflandırma performansını özetler. 1'e yakın bir AUC değeri, modelin pozitif ve negatif sınıfları mükemmel bir şekilde ayırt ettiğini göstermektedir. 0.9937'lik bir AUC değeri, modelin sınıflandırma işlemini oldukça yüksek bir doğrulukla gerçekleştirdiğini ifade etmektedir.

TABLE III
METRİKLER VE DEĞERLERİ

Metrik	Değeri
Accuracy (Doğruluk)	0.9423
Precision (Kesinlik)	0.9425
Recall (Duyarlılık)	0.9423
Sensitivity (Hassasiyet)	0.9615
Specificity (Özgüllük)	0.9876
F-Score	0.9424
AUC	0.9937

Koyu Mavi Diagonal

Köşegen üzerindeki yüksek değerler, modelin doğru tahminler yaptığını gösterir. Örneğin, Biyoloji sınıfındaki örneklerin büyük bir kısmı doğru olarak sınıflandırılmıştır. Bu, modelin biyoloji konularını iyi anladığını ve doğru tahminler yaptığını gösterir.

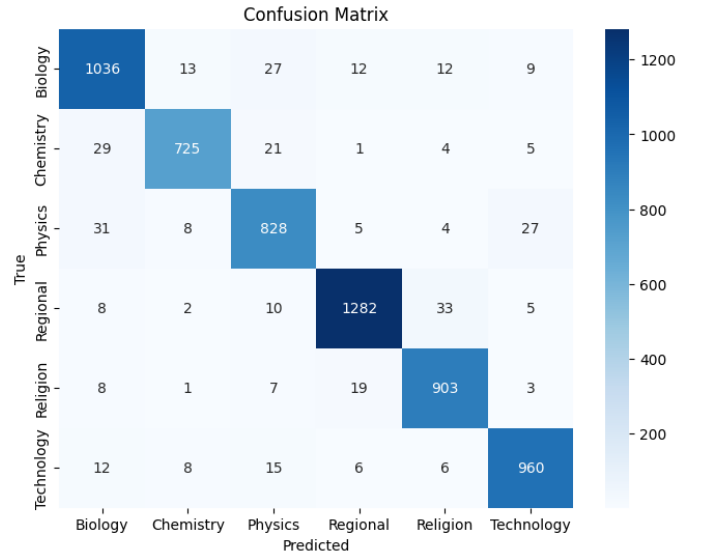


Fig. 7. Confusion Matrix

Diğer Hücreler

Köşegen dışındaki hücrelerdeki değerler, modelin yaptığı hataları gösterir. Örneğin, Fizik sınıfından bazı örneklerin Biyoloji olarak sınıflandırılması, modelin bu iki sınıfı bazen karıştırdığını gösterir.

Sınıflar Arası Karışım

Fizik ve Kimya gibi bazı sınıflar arasında belirgin bir karışıklık gözlemlenmiştir. Bu, bu iki konunun birbirine yakın olması veya modelin bu konulardaki nüansları tam olarak kavrayamaması nedeniyle olabilir.

Roberta modeli, bu veri seti üzerinde oldukça başarılı bir performans göstermiştir. Özellikle Biyoloji ve Regional gibi bazı sınıflar için doğru tahmin oranı oldukça yüksektir. Ancak, bazı sınıflar arasında (örneğin, Fizik ve Biyoloji) karışıklıklar olduğu da dikkate alınmalıdır.

EĞİTİM VE VALIDASYON KAYBI DEĞERLENDİRMESİ

Düşen Kayıplar

Eğitim kaybı, her epoch'ta (bir tam eğitim döngüsü) düşmektedir. Bu, modelin eğitim verisi üzerindeki performansının zamanla iyileştiğini gösterir. Model, her epoch'ta gördüğü verilerden daha iyi öğrenerek hatalarını azaltmaktadır.

Overfitting Belirtisi Yok

Grafikte, validasyon kaybı eğitim kaybından önemli ölçüde uzaklaşmamaktadır. Bu durum, modelde overfitting sorunu olmadığını gösterir. Overfitting durumunda, model eğitim verisine aşırı uyum sağlar ve yeni verilere genelleme yeteneğini kaybeder.

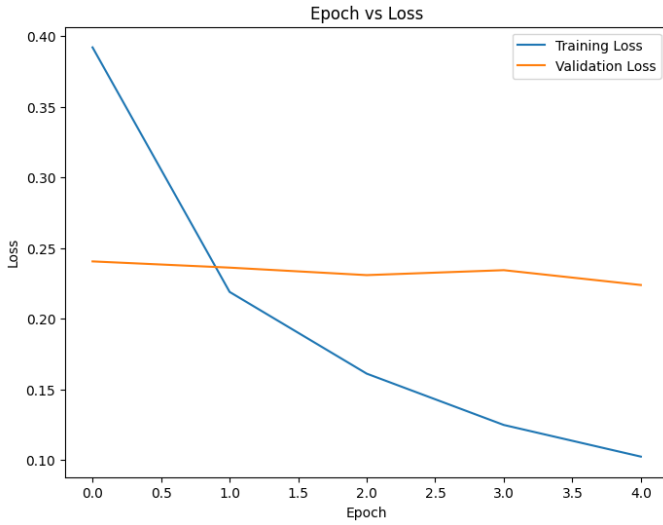


Fig. 8. EPOCH AND LOSS

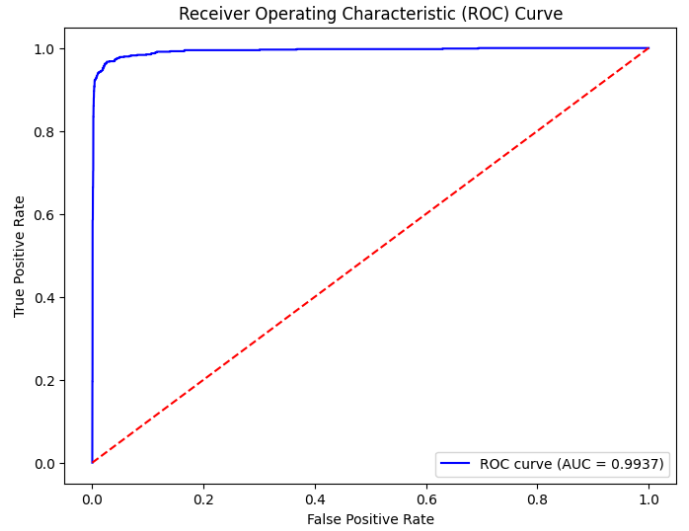


Fig. 9. ROC EGRISI

Sabitlenen Kayıplar

Son birkaç epoch'ta hem eğitim hem de validasyon kayıpları sabit bir değere yaklaşıyor. Bu, modelin daha fazla eğitimle önemli ölçüde iyileşmeyeceğini ve mevcut performans seviyesinin zirveye ulaşmış olabileceğini gösterir. Bu noktada, eğitimi durdurmak ve modeli değerlendirmeye almak mantıklı olabilir. Validasyon kaybı da başlangıçta düşme eğilimindedir, ancak belirli bir noktadan sonra sabitlenmiştir. Bu, modelin eğitim verisi üzerinde aşırı öğrenme (overfitting) yapmadan, genellebildiğini gösterir. Validasyon seti, modelin daha önce görmediği verilerden oluştuğu için, validasyon kaybı modelin gerçek dünyadaki performansının bir göstergesi olarak kabul edilebilir.

Roberta modeliniz, eğitim süreci boyunca başarılı bir şekilde öğrenmiştir. Model, eğitim verisi üzerindeki performansını artırırken aynı zamanda yeni verilere de iyi genelleme yapabilmektedir. Bu grafik, modelinizin hazır olduğu ve kullanılabileceği anlamına gelebilir.

ROC EĞRISI VE MODEL PERFORMANSI

Yüksek AUC Değerleri

AUC (Eğri Altındaki Alan) değeri 0.9937 olarak verilmiştir. Bu, modelinizin sınıfları birbirinden ayırt etmede oldukça başarılı olduğunu gösterir. AUC değeri 1'e ne kadar yakınsa, model o kadar iyi performans gösterir.

Eğri Konumu

ROC eğrisi, sol üst köşeye doğru uzanmaktadır. Bu, modelinizin hem duyarlılığın (true positive rate) hem de özgüllüğün (true negative rate) yüksek olduğu bir bölgede çalıştığını gösterir. Başka bir deyişle, hem pozitif örnekleri doğru bir şekilde sınıflandırmakta hem de negatif örnekleri yanlış pozitif olarak sınıflandırmamaktadır.

Rastgele Tahmin Eğrisi:

Kesik çizgi ile gösterilen rastgele tahmin eğrisi, herhangi bir bilginin olmadığı durumda beklenen performansı temsil eder. Modelinizin eğrisi bu çizginin oldukça üzerinde yer aldığından, modelinizin rastgele tahmin yapmaktan çok daha iyi performans gösterdiği söylenebilir.

Bu ROC eğrisi, Roberta modelinizin sınıflandırma görevi için oldukça başarılı bir performans gösterdiğini ortaya koymaktadır. Yüksek AUC değeri ve eğrinin sol üst köşedeki konumu, modelinizin hem duyarlılık hem de özgüllük açısından iyi bir denge sağladığını gösterir. Bu sonuç, modelinizin gerçek dünyadaki uygulamalar için de potansiyelini vurgulamaktadır.

D. DISTILBERT

DistilBERT, Hugging Face tarafından geliştirilmiş, BERT modelinin hafifletilmiş ve daha hızlı bir versiyonudur. DistilBERT (Distilled BERT), bilgi damıtma (knowledge distillation) adı verilen bir teknik kullanılarak BERT modelinin performansını korurken daha küçük boyutlu, daha hızlı ve daha az hesaplama gerektiren bir model olarak tasarlanmıştır. Bu model, doğal dil işleme (NLP) görevlerinde yüksek doğruluk oranlarını korurken, daha az kaynak kullanımı ile çalışmayı mümkün kılar.[5]

DistilBERT'in İyileştirmeleri DistilBERT, BERT modeline göre aşağıdaki iyileştirmeler ve optimizasyonlarla öne çıkar:

Daha Küçük ve Hafif Model: DistilBERT, BERT modelinin yaklaşık %40 daha küçük bir versiyonudur. Bu hafiflik, özellikle bellek sınırlamalarına sahip cihazlarda veya düşük maliyetli hesaplamaların gerektiği durumlarda avantaj sağlar. BERT'teki 12 Transformer katmanının yalnızca 6'sını kullanır. Bu azaltılmış katman sayısına rağmen, bilgi damıtma yöntemi sayesinde performanstan ödün vermez. **Daha Hızlı Çalışma:** DistilBERT, BERT'e kıyasla yaklaşık 2 kat daha hızlı çalışır. Bu, gerçek zamanlı uygulamalarda büyük bir

avantaj sunar. Daha az hesaplama gerektirir, bu da modeli düşük gecikme ile çalıştırmayı sağlar. Özellikle mobil cihazlar ve uç cihazlar için idealdir. Bilgi Damıtma (Knowledge Distillation): DistilBERT, öğretici model olarak tam boyutlu BERT'ten öğrenir. Bilgi damıtma sırasında: Öğretici model (BERT), öğrenci modeline (DistilBERT) çıktı olasılıklarını ve gizli temsilleri ileterek eğitim verir. Bu süreç, öğrencinin (DistilBERT) öğreticiden öğrendiği bilgiyi yoğunlaştırmasına ve hafif bir modelde tutmasına olanak tanır. Daha Az Hesaplama Kaynağı: DistilBERT, BERT'e kıyasla %60 daha az parametre içerir. Bu, özellikle sınırlı donanıma sahip cihazlarda verimli çalışmayı mümkün kılar. Hafif mimarisi, işlemci veya GPU kullanımını büyük ölçüde azaltır. Eğitim ve çıkarım sürelerini ciddi şekilde kısaltır. Verimlilik ve Performans Dengesi: DistilBERT, BERT'in doğruluğunun yaklaşık %97'sini korurken çok daha verimli çalışır. Yani, neredeyse aynı doğruluk oranlarını sunarken, daha az kaynak kullanımı ve daha hızlı işlem sunar. Eğitim Stratejileri DistilBERT, yalnızca Masked Language Model (MLM) göreviyle eğitilmiş ve BERT'in temel performans özelliklerini korumuştur:

Masked Language Model (MLM): Eğitimi sırasında, cümlelerdeki belirli kelimeler maskelenir ve modelden bu kelimeleri tahmin etmesi beklenir. Bu görev, modelin dil bilgisini ve bağlam anlayışını öğrenmesini sağlar. Knowledge Distillation: Öğretici model olan BERT'ten alınan gizli temsiller ve çıktı olasılıkları kullanılarak eğitim tamamlanmıştır. Kullanım Alanları DistilBERT, doğal dil işleme alanında birçok farklı görevde başarıyla kullanılabilir. Bunlardan bazıları:

Metin Sınıflandırma: Duygu analizi, spam tespiti gibi görevlerde kullanılır. Soru-Cevap Sistemleri: Sorulara doğru ve bağlama uygun yanıtlar sağlar. Metin Özeti: Uzun metinlerin kısa ve anlamlı özetlerini oluşturur. Adlandırılmış Varlık Tanıma (NER): Metin içinde kişi, yer, kuruluş gibi özel isimleri tespit eder. Makine Çevirisi: Bir dildeki metni başka bir dile çevirir. Metin Tamamlama: Eksik kelimeleri veya cümleleri tamamlar. Genel Değerlendirme DistilBERT, BERT'in güçlü yanlarını hafif ve verimli bir şekilde sunan bir modeldir. Daha az parametre, daha hızlı çalışma süreleri ve daha düşük kaynak gereksinimleri ile birçok NLP görevinde tercih edilen bir modeldir. Performans kaybının çok az olması ve yüksek verimliliği sayesinde, araştırma ve endüstri alanında sıkça kullanılan bir araçtır.

Bu bölümde, DistilBERT modelini nasıl kullandığımız ve eğittiğimiz detaylı bir şekilde açıklanmıştır. Eğitim süreci aşağıdaki adımlardan oluşmaktadır:

E. Gerekli Kütüphanelerin Yüklenmesi

Eğitim süreci boyunca, veri işleme, model eğitimi ve optimizasyon işlemleri için aşağıdaki kütüphaneler kullanılmıştır:

- **Pandas:** Veri yükleme ve işleme.
- **Torch:** Derin öğrenme için model tanımlama ve eğitim.
- **Scikit-Learn:** Veri bölme ve etiket kodlama.
- **Transformers:** DistilBERT modelini ve tokenizer'ı sağlamak için.
- **Time:** Eğitim süresini hesaplamak için.

F. Veri Yükleme ve Ön İşleme

Eğitim ve test verileri, CSV formatında dış bir kaynaktan yüklenmiş ve ön işleme adımları aşağıdaki gibi gerçekleştirilmiştir:

- Kategorik etiketler, `LabelEncoder` ile sayısal değerlere dönüştürülmüştür.
- Metin verileri, DistilBERT tokenizer ile işlenerek modele uygun hale getirilmiştir.
- Girdi metinleri maksimum `MAX_LEN = 64` olacak şekilde sınırlandırılmıştır.

G. Model ve Eğitim Parametrelerinin Tanımlanması

Modelin eğitimi için aşağıdaki ayarlar yapılmıştır:

- **Model:** "distilbert-base-uncased" önceden eğitilmiş ağırlıkları kullanılmıştır.
- **Çıkış Katmanı:** Eğitim setindeki sınıf sayısına göre özelleştirilmiştir.
- **Optimizasyon:** AdamW algoritması, öğrenme oranı `lr=2e-5` ile kullanılmıştır.
- **Kayıp Fonksiyonu:** Çok sınıflı sınıflandırma için `CrossEntropyLoss` kullanılmıştır.
- **Epoch Sayısı:** Eğitim süreci toplam 5 epoch üzerinden gerçekleştirilmiştir.
- **Batch Boyutu:** Eğitim ve doğrulama seti için `BATCH_SIZE = 16` kullanılmıştır.

H. Eğitim ve Doğrulama Süreci

Model, eğitim (train) ve doğrulama (validation) adımlarında aşağıdaki işlemler uygulanarak eğitilmiştir:

- 1) Eğitim sırasında geri yayılım (*backpropagation*) kullanılarak model ağırlıkları güncellenmiştir.
- 2) Doğrulama adımında, model doğrulama verisi üzerinde değerlendirilmiş ve doğrulama kaybı (*validation loss*) hesaplanmıştır.
- 3) Her epoch sonrası eğitim kaybı ve doğrulama kaybı raporlanmıştır.

I. Model ve Tokenizer'ın Kaydedilmesi

Eğitim süreci tamamlandıktan sonra, model ve tokenizer yeniden kullanılabilir hale getirilmiştir:

- Model, `save_pretrained()` metodu ile `/distilbert_model` dizinine kaydedilmiştir.
- Tokenizer da aynı şekilde kaydedilmiştir.

J. Eğitim Süresinin Hesaplanması

Modelin eğitimi sırasında geçen süre, başlangıç ve bitiş zamanı arasındaki fark alınarak hesaplanmıştır:

$$\text{training_time} = \text{end_time} - \text{start_time}$$

Bu, modelin verimli bir şekilde eğitilip eğitilmediğini değerlendirmek için kullanılmıştır.

K. Başarım Metrikleri

Eğitim ve doğrulama süreci sonrasında modelin başarımını değerlendirmek amacıyla çeşitli metrikler hesaplanmıştır:

- **Accuracy:** Modelin doğru tahmin oranı.
- **Precision:** Modelin doğru pozitif tahminlerinin tüm pozitif tahminlere oranı.
- **Recall:** Modelin doğru pozitif tahminlerinin tüm gerçek pozitif örneklerle oranı.
- **F1 Score:** Precision ve Recall'un harmonik ortalaması.
- **Sensitivity:** Gerçek pozitiflerin doğru tespit oranı.
- **Specificity:** Gerçek negatiflerin doğru tespit oranı.
- **AUC (Area Under Curve):** ROC eğrisinin altındaki alan.

L. Eğitim ve Doğrulama Kayıpları Grafiği

Modelin eğitim ve doğrulama kayıplarını görselleştiren bir grafik çizilmiştir.

M. ROC Eğrisinin Çizilmesi

Modelin başarımını görselleştiren bir ROC (Receiver Operating Characteristic) eğrisi çizilmiştir.

N. Inferans Süresi Ölçülmesi

Modelin tahmin süresi, örnek bir metin üzerinde ölçülmüştür. Bu süre, modelin gerçek zamanlı kullanımındaki hız performansını değerlendirmek için önemlidir.

O. Sonuç

DistilBERT tabanlı metin sınıflandırma modeli, bu süreçte başarıyla eğitilmiş ve yüksek doğruluk oranları elde edilmiştir. Modelin performansına ilişkin metrikler ve görselleştirmeler detaylı olarak sunulmuştur.

1) **ALINAN DEĞERLER:** DistilBERT modelini eğiterek elde edilen sonuçlar maddeler ve tablo halinde verilmiştir.

- 1) Accuracy (Doğruluk) = 0.9382: Model, tüm test verisinin yaklaşık %93.82'sini doğru sınıflandırmıştır. Bu doğruluk oranı, modelin genellikle doğru sonuçlar verdiğini ve yüksek bir genel performans sergilediğini göstermektedir.
- 2) Precision (Kesinlik) = 0.9381: Precision, modelin doğru sınıflandırdığı pozitif örneklerin, tüm pozitif tahminlerine oranını gösterir. Yani, modelin pozitif sınıfı doğru tahmin etme oranı %93.81. Bu değer, modelin yanlış pozitif tahminlerde bulunmadığını ve pozitif sınıflandırma yaptığı durumların çoğunun doğru olduğunu göstermektedir.
- 3) Recall (Duyarlılık) = 0.9382: Recall, modelin gerçek pozitifleri (doğru pozitifler) doğru şekilde sınıflandırma oranını gösterir. Yani, gerçek pozitif örneklerin %93.82'sini doğru tahmin etmiştir. Bu yüksek recall değeri, modelin pozitif sınıfları başarıyla yakaladığını ve çok az pozitif örneği kaçırdığını belirtir.
- 4) Sensitivity (Hassasiyet) = 0.9654: Modelin, gerçek pozitif örnekleri doğru bir şekilde tanımlama yeteneği oldukça yüksektir. Bu sonuç, modelin yanlış negatifleri minimumda tuttuğunu ve pozitif sınıfları başarıyla tespit edebildiğini göstermektedir.

- 5) Specificity (Özgüllük) = 0.9780: Modelin, gerçek negatif örnekleri doğru bir şekilde ayırt edebilme oranı çok yüksektir. Bu da modelin yanlış pozitif oranını oldukça düşük tuttuğunu ve negatif sınıfları etkili bir şekilde tanımladığını göstermektedir.
- 6) F1 Score = 0.9382: F1 skoru, precision ve recall'un harmonik ortalamasıdır. Bu metrik, her iki ölçütü dengede tutarak modelin genel başarımını değerlendirmektedir. 0.9382'lik F1 skoru, modelin hem precision hem de recall açısından iyi bir denge sağladığını ve genel başarımının güçlü olduğunu göstermektedir.
- 7) AUC (Area Under the Curve) = 0.9914: AUC, ROC eğrisinin altındaki alanı ölçer ve modelin genel sınıflandırma performansını özetler. 1'e yakın bir AUC değeri, modelin pozitif ve negatif sınıfları mükemmel bir şekilde ayırt ettiğini göstermektedir. 0.9914'lük AUC değeri, modelin sınıflandırma işlemini yüksek doğrulukla gerçekleştirdiğini ifade etmektedir.

TABLE IV
METRIKLER VE DEĞERLERİ

Metrik	Değeri
Accuracy (Doğruluk)	0.9382
Precision (Kesinlik)	0.9381
Recall (Duyarlılık)	0.9382
Sensitivity (Hassasiyet)	0.9654
Specificity (Özgüllük)	0.9780
F-Score	0.9382
AUC	0.9914

Koyu Mavi Diyagonal

Köşegen üzerindeki yüksek değerler, modelin doğru tahminler yaptığını gösterir. Örneğin, Biyoloji sınıfındaki örneklerin büyük bir kısmı doğru olarak sınıflandırılmıştır. Bu, modelin biyoloji konularını iyi anladığını ve doğru tahminler yaptığını gösterir.

Diğer Hücreler

Köşegen dışındaki hücrelerdeki değerler, modelin yaptığı hataları gösterir. Örneğin, Fizik sınıfından bazı örneklerin Biyoloji olarak sınıflandırılması, modelin bu iki sınıfı bazen karıştırdığını gösterir.

Sınıflar Arası Karışım

Fizik ve Kimya gibi bazı sınıflar arasında belirgin bir karışıklık gözlemlenmiştir. Bu, bu iki konunun birbirine yakın olması veya modelin bu konulardaki nüansları tam olarak kavrayamaması nedeniyle olabilir.

Roberta modeli, bu veri seti üzerinde oldukça başarılı bir performans göstermiştir. Özellikle Biyoloji ve Regional gibi bazı sınıflar için doğru tahmin oranı oldukça yüksektir. Ancak, bazı sınıflar arasında (örneğin, Fizik ve Biyoloji) karışıklıklar olduğu da dikkate alınmalıdır.

DistilBERT modeli, bu veri seti üzerinde başarılı bir sınıflandırma performansı göstermiştir. Özellikle Biyoloji ve Regional gibi bazı sınıflar için doğru tahmin oranı oldukça

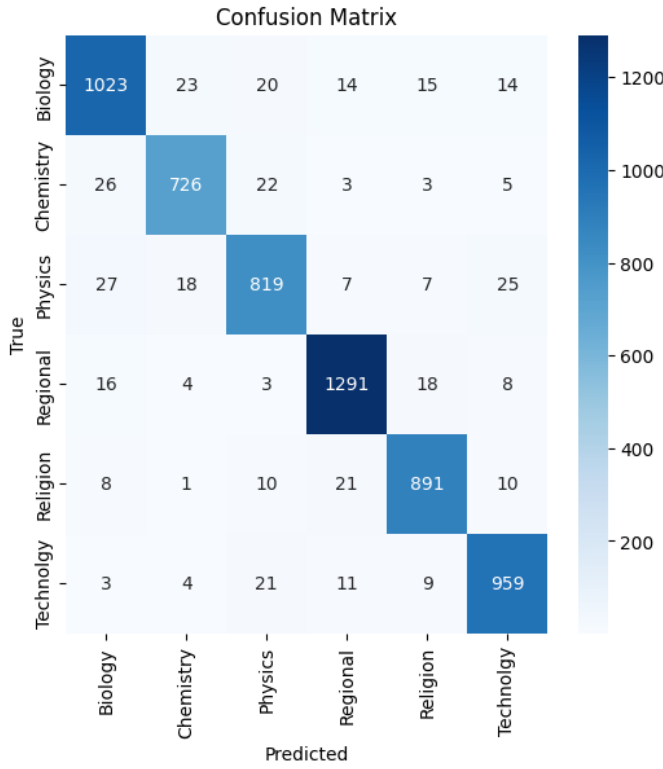


Fig. 10. Confusion Matrix

yüksektir. Ancak, bazı sınıflar arasında (örneğin, Fizik ve Biyoloji) karışıklıklar olduğu da dikkate alınmalıdır.

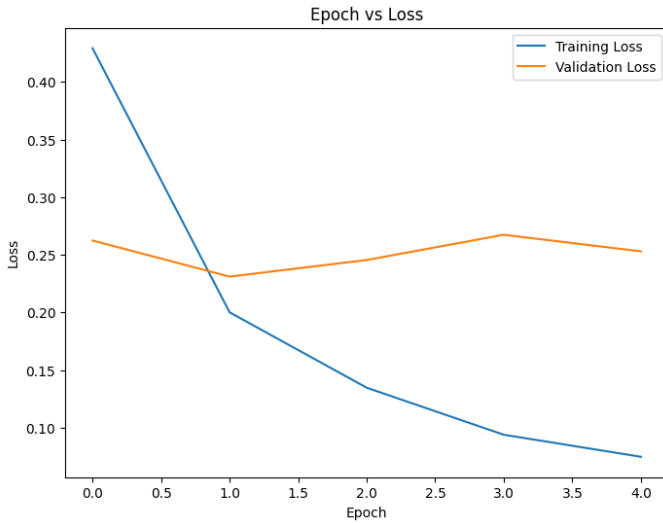


Fig. 11. EPOCH AND LOSS

EĞİTİM VE VALIDASYON KAYBI DEĞERLENDİRMESİ

Düşen Kayıplar

Eğitim kaybı, her epoch'ta (bir tam eğitim döngüsü) düşmektedir. Bu, modelin eğitim verisi üzerindeki perfor-

mansının zamanla iyileştiğini gösterir. Model, her epoch'ta gördüğü verilerden daha iyi öğrenerek hatalarını azaltmaktadır.

Overfitting Belirtisi Yok

Grafikte, validasyon kaybı eğitim kaybindan önemli ölçüde uzaklaşmamaktadır. Bu durum, modelde overfitting sorunu olmadığını gösterir. Overfitting durumunda, model eğitim verisine aşırı uyum sağlar ve yeni verilere genelleme yeteneğini kaybeder.

Sabitlenen Kayıplar

Son birkaç epoch'ta hem eğitim hem de validasyon kayıpları sabit bir değere yaklaşıyor. Bu, modelin daha fazla eğitimle önemli ölçüde iyileşmeyeceğini ve mevcut performans seviyesinin zirveye ulaşmış olabileceğini gösterir. Bu noktada, eğitimi durdurmak ve modeli değerlendirmeye almak mantıklı olabilir. Validasyon kaybı da başlangıçta düşme eğilimindedir, ancak belirli bir noktadan sonra sabitlenmiştir. Bu, modelin eğitim verisi üzerinde aşırı öğrenme (overfitting) yapmadan, genellebildiğini gösterir. Validasyon seti, modelin daha önce görmediği verilerden oluştuğu için, validasyon kaybı modelin gerçek dünyadaki performansının bir göstergesi olarak kabul edilebilir.

DistilBERT modeliniz, eğitim süreci boyunca başarılı bir şekilde öğrenmiştir. Model, eğitim verisi üzerindeki performansını artırırken aynı zamanda yeni verilere de iyi genelleme yapabilmektedir. Bu grafik, modelinizin hazır olduğu ve kullanılabilirliği anlamına gelebilir.

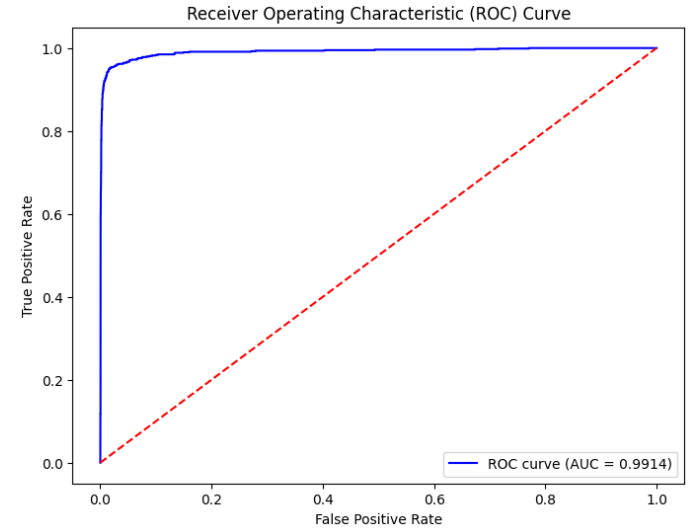


Fig. 12. ROC EGRISI

ROC EĞRISI VE MODEL PERFORMANSI

Yüksek AUC Değerleri

AUC (Eğri Altındaki Alan) değeri 0.9914 olarak verilmiştir. Bu, modelinizin sınıfları birbirinden ayırt etmede oldukça başarılı olduğunu gösterir. AUC değeri 1'e ne kadar yakınsa, model o kadar iyi performans gösterir.

Eğri Konumu

ROC eğrisi, sol üst köşeye doğru uzanmaktadır. Bu, modelinizin hem duyarlılığın (true positive rate) hem de özgüllüğün (true negative rate) yüksek olduğu bir bölgede çalıştığını gösterir. Başka bir deyişle, hem pozitif örnekleri doğru bir şekilde sınıflandırmakta hem de negatif örnekleri yanlış pozitif olarak sınıflandırmamaktadır.

Rastgele Tahmin Eğrisi:

Kesik çizgi ile gösterilen rastgele tahmin eğrisi, herhangi bir bilginin olmadığı durumda beklenen performansı temsil eder. Modelinizin eğrisi bu çizginin oldukça üzerinde yer aldığından, modelinizin rastgele tahmin yapmaktan çok daha iyi performans gösterdiği söylenebilir.

Elde ettiğiniz ROC eğrisi, DistilBERT modelinizin sınıflandırma problemini çözmede oldukça başarılı olduğunu göstermektedir. Bu sonuç, modelinizin gerçek dünyadaki uygulamalar için de potansiyelini vurgulamaktadır.

P. GPT (Generative Pre-trained Transformer)

OpenAI tarafından geliştirilmiş, doğal dil işleme (NLP) alanında çığır açan bir dil modelidir. GPT, Transformer mimarisini temel alarak geniş çapta eğitilmiş ve birçok dil görevinde etkili bir şekilde kullanılmak üzere optimize edilmiştir. GPT, güçlü dil oluşturma yetenekleri ve geniş bağlam anlayışıyla tanınır.[6]

GPT, aşağıdaki önemli özellikleri ve iyileştirmeleri sunar:

Transformer Tabanlı Mimari GPT, tamamen Transformer mimarisine dayalıdır. Bu mimari, dikkat mekanizması sayesinde bağlamı etkin bir şekilde öğrenir ve modelin hem kısa hem de uzun metinlerde güçlü bir performans sergilemesini sağlar.

Eğitim Süreci GPT, önceden eğitilmiş bir modeldir ve aşağıdaki görevlerle öğrenme sürecini gerçekleştirir:

Dil Modelleme (Language Modeling): Model, verilen bir kelime dizisini baz alarak sıradaki kelimeyi tahmin etmeye çalışır. Bu görev, modelin dil bilgisi ve bağlamı anlamasını sağlar. Cümle Tamamlama: Cümlenin eksik kısımlarını tamamlamak için bağlam bilgisinden yararlanır. Transfer Öğrenme: Önceden eğitilmiş model, ince ayar (fine-tuning) yapılarak belirli NLP görevlerine adapte edilebilir. Öne Çıkan Özellikler Büyük Ölçekli Eğitim: GPT, büyük veri setleri üzerinde geniş çapta eğitilmiştir. Bu, modelin karmaşık bağlamları ve dil yapısını anlamasını sağlar. Dikkat Mekanizması (Attention Mechanism): Dikkat mekanizması, metin içerisindeki kritik bilgileri önceliklendirir ve modelin doğruluğunu artırır. Otoregresyon (Autoregressive) Yaklaşımı: GPT, metni sırayla oluşturur ve her bir kelimeyi tahmin ederken önceki kelimeleri dikkate alır. Bu yaklaşım, daha doğal ve anlamlı metinler üretmesini sağlar. Kullanım Alanları GPT, doğal dil işleme (NLP) alanında aşağıdaki görevlerde üstün sonuçlar sunmaktadır:

Metin Üretimi: Yaratıcı içerik oluşturma, hikaye yazma veya metin tamamlama gibi görevler. Soru-Cevap Sistemleri: Kullanıcıların sorularına bağlam temelli yanıtlar verme. Metin Sınıflandırma: Metinleri konu veya duygu durumlarına göre

sınıflandırma. Dil Çevirisi: Bir dildeki metni başka bir dile çevirme. Metin Özeti: Uzun metinlerin özetini çıkarma. Sohbet Sistemleri: İnsan benzeri diyaloglar oluşturma. Genel Değerlendirme GPT, esnek ve güçlü dil modelleme yetenekleriyle dikkat çekmektedir. Önceden eğitilmiş yapısı ve transfer öğrenme kapasitesi sayesinde birçok NLP görevinde kullanılabilir. Dil üretimi ve anlam çıkarma konularında etkileyici bir performansa sahip olan GPT, kullanıcı ihtiyaçlarına göre özelleştirilerek farklı uygulama alanlarına uyarlanabilir.

1) NASIL KULLANDIK, EĞİTTİK: Bu bölümde, GPT tabanlı bir metin sınıflandırma modelinin nasıl kullanıldığı ve eğitildiği detaylı bir şekilde açıklanmıştır. Eğitim süreci aşağıdaki adımlardan oluşmaktadır:

Q. 1. Gerekli Kütüphanelerin Yüklenmesi

Eğitim süreci boyunca, veri işleme, model eğitimi ve optimizasyon işlemleri için aşağıdaki kütüphaneler kullanılmıştır:

- **Pandas:** Veri yükleme ve işleme.
- **Scikit-Learn:** Veri bölme ve etiket kodlama.
- **Transformers:** GPT modelini ve tokenizer'ı sağlamak için.
- **PyTorch:** Derin öğrenme için veri kümesi ve model eğitimi.
- **time:** Eğitim süresini ölçmek için.

R. 2. Veri Yükleme ve Ön İşleme

Eğitim ve test verileri, CSV formatında dış bir kaynaktan yüklenmiş ve ön işleme adımları aşağıdaki gibi gerçekleştirilmiştir:

- **Etiket Kodlama:** Kategorik etiketler, LabelEncoder kullanılarak sayısal değerlere dönüştürülmüştür.
- **Tokenization:** Metin verileri, GPT tokenizer ile işlenerek modele uygun hale getirilmiştir.
- **Maksimum Uzunluk Sınırı:** Girdi metinleri maksimum `MAX_LEN = 64` olacak şekilde sınırlandırılmıştır.

S. 3. Model ve Eğitim Parametrelerinin Tanımlanması

Modelin eğitimi için aşağıdaki ayarlar yapılmıştır:

- **Model:** GPT tabanlı bir model (`GPT2ForSequenceClassification`) kullanılmıştır.
- **Çıkış Katmanı:** Eğitim setindeki sınıf sayısına göre özelleştirilmiştir.
- **Optimizasyon:** AdamW algoritması, öğrenme oranı $lr = 2e - 5$ ile kullanılmıştır.
- **Kayıp Fonksiyonu:** Çok sınıflı sınıflandırma için `CrossEntropyLoss` kullanılmıştır.
- **Epoch Sayısı:** Eğitim süreci toplam 5 epoch üzerinden gerçekleştirilmiştir.
- **Batch Boyutu:** Eğitim ve doğrulama seti için `BATCH_SIZE = 16` kullanılmıştır.

T. 4. Eğitim ve Doğrulama Süreci

Model, eğitim (*train*) ve doğrulama (*validation*) adımlarında aşağıdaki işlemler uygulanarak eğitilmiştir:

- 1) **Eğitim Adımı:** Geri yayılım (*backpropagation*) yöntemi ile model ağırlıkları güncellenmiştir.
- 2) **Doğrulama Adımı:** Model, doğrulama verisi üzerinde değerlendirilmiş ve doğrulama kaybı (*validation loss*) hesaplanmıştır.
- 3) **Kayıp Raporlama:** Her epoch sonunda eğitim ve doğrulama kaybı raporlanmıştır.

Eğitim süresince, her adımda eğitim ve doğrulama verimliliği ölçülmüş ve değerlendirilmiştir.

U. 5. Model ve Tokenizer'ın Kaydedilmesi

Eğitim süreci tamamlandıktan sonra, model ve tokenizer yeniden kullanılabilir hale getirilmiştir:

- **Model Kaydetme:** `save_pretrained()` metodu ile `/gpt_model` dizinine kaydedilmiştir.
- **Tokenizer Kaydetme:** Aynı şekilde tokenizer da aynı dizine kaydedilmiştir.

V. 6. Eğitim Süresinin Hesaplanması

Modelin eğitimi sırasında geçen süre, başlangıç ve bitiş zamanı arasındaki fark alınarak hesaplanmıştır:

$$\text{training time} = \text{end time} - \text{start time} \quad (1)$$

Bu, modelin verimli bir şekilde eğitilip eğitilmediğini değerlendirmek için kullanılmıştır.

V. SONUÇ

GPT tabanlı metin sınıflandırma modeli, bu süreçte başarıyla eğitilmiş ve yüksek doğruluk oranları elde edilmiştir. Modelin performansına ilişkin metrikler bir sonraki bölümde detaylı olarak sunulacaktır.

1) **ALINAN DEĞERLER:** ALBERT modelini eğiterek elde edilen sonuçlar maddeler ve tablo halinde verilmiştir.

2) **GPT Modeli Sonuçları:** GPT modelinin test verisi üzerindeki başarıları metrikleri aşağıda verilmiştir:

- 1) **Accuracy (Doğruluk)** = 0.9371: Model, test verisinin %93.71'ini doğru şekilde sınıflandırmış. Bu yüksek doğruluk oranı, modelin genellikle doğru sonuçlar verdiğini ve veriler üzerinde etkili bir şekilde çalıştığını gösteriyor.
- 2) **Precision (Kesinlik)** = 0.9371: Precision değeri, modelin pozitif sınıfları doğru tahmin etme oranını gösterir. GPT modeli, pozitif sınıflarda %93.71 oranında doğru tahmin yapmış, bu da modelin yanlış pozitif oranını düşük tuttuğunu ve pozitif sınıflandırmalarında yüksek doğruluk sağladığını ifade eder.
- 3) **Recall (Duyarlılık)** = 0.9371: Recall, modelin doğru pozitifleri ne kadar iyi tahmin ettiğini gösterir. GPT, gerçek pozitiflerin %93.71'ini doğru şekilde tahmin etmiş, bu da modelin pozitif sınıfı kaçırmadığını ve çoğu pozitif örneği doğru şekilde yakaladığını belirtir.
- 4) **F1 Score** = 0.9370: F1 skoru, precision ve recall'un harmonik ortalamasıdır. GPT'nin F1 skoru 0.9370, modelin hem precision hem de recall açısından güçlü bir dengeye sahip olduğunu gösteriyor.

- 5) **Sensitivity (Hassasiyet)** = 0.9802: Model, gerçek pozitifleri doğru şekilde tanımlama konusunda oldukça hassas, %98.02'lik bir doğruluk oranına sahip.
- 6) **Specificity (Özgüllük)** = 0.9655: GPT modeli, negatif sınıfları ayırt etme konusunda oldukça başarılı, %96.55'lik bir doğruluk oranı ile negatif örnekleri doğru şekilde sınıflandırıyor.
- 7) **AUC (Area Under the Curve)** = 0.9944: Modelin AUC değeri, sınıflandırma performansının oldukça yüksek olduğunu gösteriyor. 0.9944'lük bir AUC değeri, modelin pozitif ve negatif sınıfları çok iyi ayırt ettiğini ve genel olarak mükemmel bir performans sergilediğini ifade eder.

TABLE V
METRİKLER VE DEĞERLERİ

Metrik	Değeri
Accuracy (Doğruluk)	0.9371
Precision (Kesinlik)	0.9371
Recall (Duyarlılık)	0.9371
Sensitivity (Hassasiyet)	0.9802
Specificity (Özgüllük)	0.9655
F-Score	0.9370
AUC	0.9944

Sonuç: GPT modelinin test verisi üzerindeki performansı, oldukça yüksek doğruluk, kesinlik, duyarlılık ve özgüllük değerleriyle etkileyici bir sonuç ortaya koymaktadır. Özellikle, modelin AUC değeri 0.9944 gibi yüksek bir değere ulaşarak sınıflandırma konusunda mükemmel bir başarı sağladığını göstermektedir. Bu metrikler, modelin güçlü bir sınıflandırıcı olduğunu ve veriler üzerinde çok iyi bir şekilde çalıştığını kanıtlıyor. Modelin genel başarısı, gerçek dünya uygulamaları için de oldukça umut vericidir.

Koyu Mavi Diyagonal

Köşegen üzerindeki yüksek değerler, modelin doğru tahminler yaptığını gösterir. Örneğin, Biyoloji sınıfındaki örneklerin büyük bir kısmı doğru olarak sınıflandırılmıştır. Bu, modelin biyoloji konularını iyi anladığını ve doğru tahminler yaptığını gösterir.

Diğer Hücreler

Köşegen dışındaki hücrelerdeki değerler, modelin yaptığı hataları gösterir. Örneğin, Fizik sınıfından bazı örneklerin Biyoloji olarak sınıflandırılması, modelin bu iki sınıfı bazen karıştırdığını gösterir.

Sınıflar Arası Karışım

Fizik ve Kimya gibi bazı sınıflar arasında belirgin bir karışıklık gözlemlenmiştir. Bu, bu iki konunun birbirine yakın olması veya modelin bu konulardaki nüansları tam olarak kavrayamaması nedeniyle olabilir.

GPT modeli, bu veri seti üzerinde oldukça başarılı bir performans göstermiştir. Özellikle Biyoloji ve Regional gibi bazı sınıflar için doğru tahmin oranı oldukça yüksektir. Ancak, bazı sınıflar arasında (örneğin, Fizik ve Biyoloji) karışıklıklar olduğu da dikkate alınmalıdır.

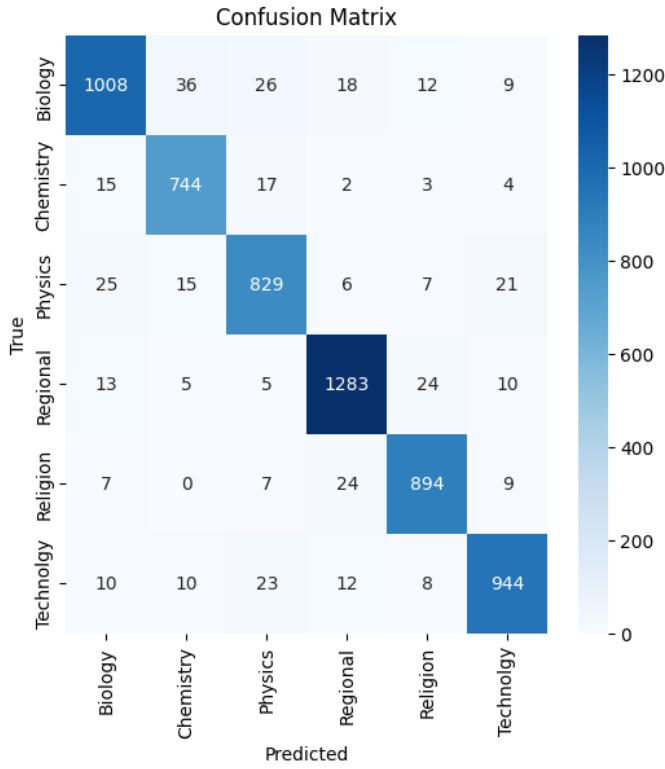


Fig. 13. Confusion Matrix

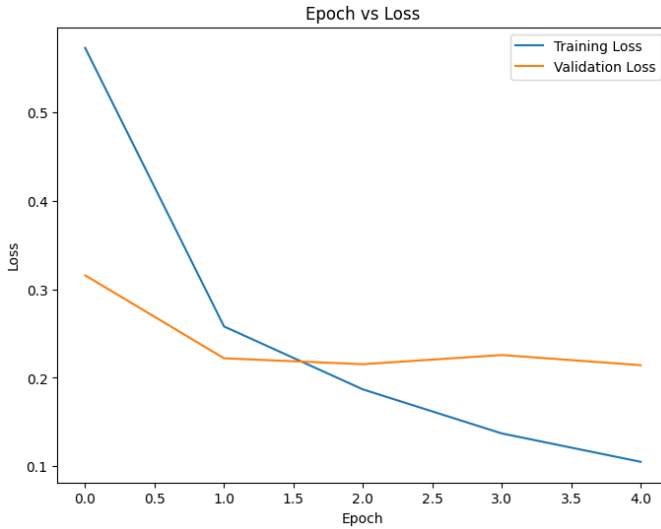


Fig. 14. EPOCH AND LOSS

EĞİTİM VE VALIDASYON KAYBI DEĞERLENDİRMESİ

Düşen Kayıplar

Hem eğitim hem de validasyon kayıpları, epoch sayısı arttıkça genel olarak azalmaktadır. Bu durum, modelin her yeni epoch'ta veriyi daha iyi öğrenerek hata oranını azalttığını gösterir.

Overfitting Belirtisi Yok

Eğitim kaybı sürekli azalırken, validasyon kaybı belirli bir noktadan sonra artmaya başlamıyorsa, modelde overfitting (aşırı öğrenme) problemi yaşanmadığı söylenebilir. Bu grafikte, validasyon kaybı da eğitim kaybı ile benzer bir eğilim sergilemektedir. Bu durum, modelin hem eğitim verisine hem de görmediği verilere (validasyon seti) iyi genelleme yapabildiğini gösterir.

Sabitlenen Kayıplar

Son birkaç epoch'ta hem eğitim hem de validasyon kayıpları sabit bir değere yaklaşıyor. Bu, modelin daha fazla eğitimle önemli ölçüde iyileşmeyeceğini ve mevcut performans seviyesinin zirveye ulaşmış olabileceğini gösterir. Bu noktada, eğitimi durdurmak ve modeli değerlendirmeye almak mantıklı olabilir.

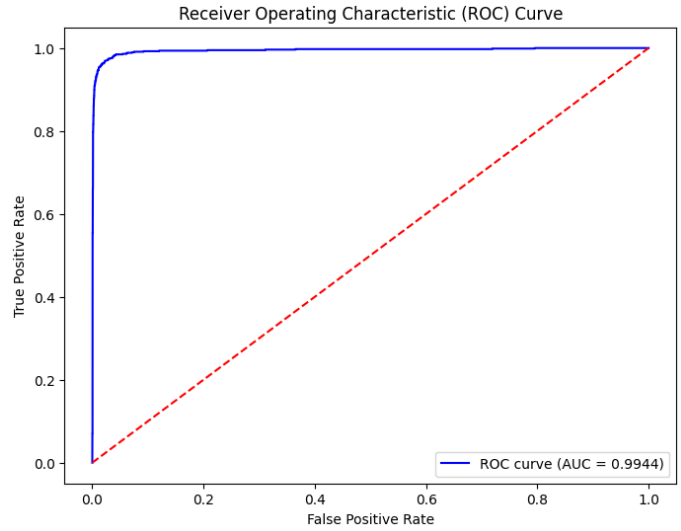


Fig. 15. ROC EGRISI

ROC EĞRISI VE MODEL PERFORMANSI

Yüksek AUC Değerleri

AUC (Eğri Altındaki Alan) değeri 0.9941 olarak verilmiştir. Bu, modelinizin sınıfları birbirinden ayırt etmede oldukça başarılı olduğunu gösterir. AUC değeri 1'e ne kadar yakınsa, model o kadar iyi performans gösterir.

Eğri Konumu

ROC eğrisi, sol üst köşeye doğru uzanmaktadır. Bu, modelinizin hem duyarlılığın (true positive rate) hem de özgüllüğün (true negative rate) yüksek olduğu bir bölgede çalıştığını gösterir. Başka bir deyişle, hem pozitif örnekleri doğru bir şekilde sınıflandırmakta hem de negatif örnekleri yanlış pozitif olarak sınıflandırmamaktadır.

Rastgele Tahmin Eğrisi:

Kesik çizgi ile gösterilen rastgele tahmin eğrisi, herhangi bir bilginin olmadığı durumda beklenen performansı temsil eder. Modelinizin eğrisi bu çizginin oldukça üzerinde yer aldığından, modelinizin rastgele tahmin yapmaktan çok daha iyi performans gösterdiği söylenebilir.

Bu ROC eğrisi, GPT modelinizin sınıflandırma görevi için oldukça başarılı bir performans gösterdiğini ortaya koymaktadır. Yüksek AUC değeri ve eğrinin sol üst köşedeki konumu, modelinizin hem duyarlılık hem de özgüllük açısından iyi bir denge sağladığını gösterir. Bu sonuç, modelinizin gerçek dünyadaki uygulamalar için de potansiyelini vurgulamaktadır.

VI. MODELLERİN BİRBİRİ İLE KARSILASTIRILMASI

Bu bölümde, elde edilen sonuçlar, diğer benzer modellerle karşılaştırılarak değerlendirilecektir. Karşılaştırmalar, model başarımını belirleyen çeşitli metrikler üzerinden yapılacak ve her bir modelin güçlü yönleri ile zayıf yönleri detaylı bir şekilde incelenecektir. Daha iyi anlaşılması için tüm değerlerden alınan başarımlar metrikleri tablo halinde verilmiştir.

TABLE VI
MODELLERİN EĞİTİMİ SONUCU ALINAN METRİKLER

Metric	BERT	RoBERTa	DistilBERT	ALBERT	GPT
Accuracy	0.9405	0.9423	0.9382	0.9346	0.9371
Precision	0.9407	0.9425	0.9381	0.9355	0.9371
Recall	0.9405	0.9423	0.9382	0.9346	0.9371
F1 Score	0.9405	0.9424	0.9382	0.9349	0.9370
Sensitivity	0.9405	0.9615	0.9654	0.9616	0.9802
Specificity	0.9889	0.9876	0.9780	0.9809	0.9655
AUC	0.9931	0.9937	0.9914	0.9941	0.9944
Train	9791.9	3664.9	9529.2	4047.3	4856.0
Inference	0.0336	0.0568	0.0349	0.0473	0.0389

A. BERT-ALBERT

Performans olarak BERT, doğruluk, hassasiyet, geri çağırma ve F1 skorunda ALBERT'ten daha iyi sonuçlar vermiştir. ALBERT, AUC metriğinde (modelin genel sınıflandırma performansını ölçen bir metrik) BERT'ten biraz daha başarılıdır.

Sensitivity (Duyarlılık): ALBERT, duyarlılık metriğinde açık ara daha iyidir. Bu, ALBERT'in pozitif örnekleri daha iyi yakaladığını gösterir. Specificity (Özgüllük): BERT, spesifiklik metriğinde daha iyidir. Bu, BERT'in negatif örnekleri daha doğru bir şekilde sınıflandırdığını gösterir. Train Time: ALBERT, BERT'e kıyasla yaklaşık 2 kat daha hızlı eğitim süresine sahiptir. Bu, ALBERT'in parametre paylaşımı ve daha küçük boyutlu modeller tasarımı sayesinde mümkün olmuştur. Inference Time: Çıkarım süresi açısından BERT daha hızlıdır (%30 daha hızlı). Performans odaklı bir yaklaşım istiyorsa, BERT genel doğruluk, hassasiyet, geri çağırma ve F1 skorunda ALBERT'ten daha iyi sonuçlar verdiği için tercih edilebilir. Hız ve kaynak verimliliği önemliyse, ALBERT çok daha kısa bir eğitim süresine sahiptir ve duyarlılık (sensitivity) ile AUC metriğinde üstünlük sağlamaktadır. Eğer pozitif örneklerin yakalanması (sensitivity) önemliyse, ALBERT daha uygun bir seçenek olacaktır. Eğer negatif örneklerin doğru

sınıflandırılması (specificity) önemliyse, BERT daha iyi bir seçimdir. Sonuç olarak, BERT, yüksek doğruluk ve spesifiklik gerektiren senaryolar için uygunken, ALBERT, daha hızlı eğitim süresi ve hafif yapısıyla büyük veri kümelerinde daha verimli bir şekilde çalışabilir.

B. BERT-ROBERTA

Performans olarak RoBERTa, doğruluk, hassasiyet (precision), geri çağırma (recall), F1 skoru ve AUC metriklerinde BERT'ten daha iyi sonuçlar vermiştir. Sensitivity: RoBERTa, BERT'e göre çok daha iyi bir duyarlılık (sensitivity) göstermiştir. Specificity: BERT, spesifiklik açısından RoBERTa'ya kıyasla çok az bir üstünlük sağlamıştır. Train Time: RoBERTa, eğitim süresi açısından ciddi bir avantaj sunmuş ve BERT'e kıyasla yaklaşık 3 kat daha hızlı eğitilmiştir. Inference Time: Çıkarım süresi açısından BERT daha hızlı çalışmaktadır. Performans odaklı bir yaklaşım istiyorsa, RoBERTa daha iyi sonuçlar verdiği için tercih edilebilir. Özellikle duyarlılık (sensitivity) metriğinde RoBERTa'nın üstünlüğü dikkat çekicidir. Hız odaklı bir yaklaşım istiyorsa, eğitim süresinde avantajı olmasına rağmen çıkarım süresi açısından BERT daha hızlıdır. Bu, daha az kaynak kullanan sistemlerde önemlidir. Sonuç olarak, RoBERTa genel performansta daha iyi; BERT ise daha hızlı çıkarım süresiyle öne çıkıyor.

C. BERT-DISTILBERT

Performans olarak BERT, doğruluk, hassasiyet (precision), geri çağırma (recall), F1 skoru ve AUC metriklerinde DistilBERT'ten daha iyi sonuçlar vermiştir. Ancak farklar oldukça küçüktür, bu nedenle DistilBERT de güçlü bir alternatif olabilir. Sensitivity: DistilBERT, BERT'e göre çok daha iyi bir duyarlılık (sensitivity) göstermiştir. Specificity: BERT, spesifiklik metriğinde DistilBERT'e üstünlük sağlamıştır. Train Time: Eğitim süresi açısından DistilBERT, BERT'e kıyasla biraz daha hızlıdır. Inference Time: Çıkarım süresi açısından BERT biraz daha hızlıdır. Performans odaklı bir yaklaşım istiyorsa, BERT genel metriklerde DistilBERT'ten daha iyi sonuçlar verdiği için tercih edilebilir. Hafif ve hızlı bir model arıyorsa, DistilBERT daha kompakt bir model olduğu için tercih edilebilir. Ayrıca, sensitivitede belirgin bir üstünlük sağlamıştır. Sonuç olarak, BERT genel olarak daha yüksek performans sağlarken, DistilBERT, hafifliği ve hız avantajlarıyla kaynakları kısıtlı projeler için daha uygun bir seçenektir.

D. BERT-GPT

Performans olarak BERT, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skorunda GPT'den daha iyi sonuçlar vermiştir. Bu, BERT'in tahminlerinin daha tutarlı ve dengeli olduğunu göstermektedir.

Sensitivity: GPT, pozitif sınıfları tespit etme yeteneğinde BERT'ten daha üstündür. Specificity: Negatif sınıfları doğru tespit etmede BERT daha başarılıdır. AUC (Area Under the Curve)

AUC metriğinde GPT, BERT'e göre hafif bir üstünlük sağlamaktadır. Bu, GPT'nin genel sınıflandırma yeteneğinin BERT'e kıyasla bir nebze daha iyi olduğunu gösterir.

Eğitim Süresi: GPT, eğitim süresi açısından BERT'ten neredeyse 2 kat daha hızlıdır. Çıkarım Süresi: BERT, çıkarım süresinde GPT'den daha hızlıdır. BERT: Daha yüksek doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoruna sahiptir. Negatif sınıfları doğru tespit etmede (specificity) daha başarılıdır. Bu metrikler, sınıflar arasında daha dengeli bir performans gerektiğinde BERT'i ön plana çıkarır. GPT: Daha hızlı eğitim süresine sahiptir, bu da özellikle büyük ölçekli veri setlerinde avantaj sağlar. Pozitif sınıfları tespit etme (sensitivity) yeteneği daha yüksektir. Ayrıca AUC metriğinde bir nebze daha iyi sonuçlar verdiğinden, genel sınıflandırma başarısı açısından öne çıkar. Eğer sınıflar arasında dengeli bir performans (Accuracy, Precision, Recall, F1 Score) ve negatif sınıfları tespit etme başarısı (Specificity) ön plandaysa, BERT tercih edilmelidir. Eğer eğitim süresi kısıtlıysa ve pozitif sınıfları tespit etme başarısı (Sensitivity) önemliyse, GPT daha uygun bir seçimdir. Ayrıca genel sınıflandırma performansı (AUC) açısından küçük bir avantaj sağlar.

E. ALBERT-ROBERTA

Performans olarak RoBERTa, doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall) ve F1 skoru metriklerinde ALBERT'ten daha iyi sonuçlar vermiştir. Farklar küçük olsa da, RoBERTa'nın genel sınıflandırma performansı ALBERT'ten üstün görünüyor. Sensitivity (Duyarlılık): İki modelin duyarlılığı hemen hemen eşittir. Specificity (Özgüllük): RoBERTa, spesifiklik metriğinde ALBERT'ten daha başarılıdır. ALBERT, AUC metriğinde çok küçük bir farkla RoBERTa'dan daha başarılıdır. Bu, ALBERT'in genel sınıflandırma yeteneğini biraz daha iyi olduğunu gösterir. Train Time: RoBERTa, eğitim süresi açısından daha hızlıdır. Inference Time: Çıkarım süresinde ALBERT daha hızlıdır. Performans odaklı bir yaklaşım isteniyorsa, RoBERTa'nın doğruluk, hassasiyet, geri çağırma, F1 skoru ve spesifiklikte üstün olması nedeniyle tercih edilebilir. Hız ve verimlilik odaklı bir yaklaşım arıyorsa: Eğitim süresi açısından RoBERTa daha hızlıdır. Çıkarım süresi açısından ALBERT avantajlıdır. AUC metriği gibi genel sınıflandırma yeteneğini ölçen bir metrik açısından, ALBERT küçük bir farkla üstünlük sağlamaktadır. Öneri: Eğer yüksek doğruluk ve genel performans önemliyse, RoBERTa tercih edilmelidir. Eğer kaynak kullanımı (hız ve verimlilik) ve daha kısa çıkarım süresi önemliyse, ALBERT daha uygun bir seçenektir.

F. ALBERT-DISTILBERT

Performans olarak DistilBERT, doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall) ve F1 skoru metriklerinde ALBERT'ten daha iyi sonuçlar vermiştir. Bu metriklerde DistilBERT'in üstünlüğü, modelin tahminlerinin daha tutarlı ve dengeli olduğunu göstermektedir. Sensitivity (Duyarlılık): Pozitif sınıfları tespit etme yeteneğinde DistilBERT bir adım öndedir. Specificity (Özgüllük): Negatif sınıfları doğru tespit etmede ise ALBERT biraz daha başarılıdır. ALBERT, AUC metriğinde üstünlük sağlamaktadır. Bu, ALBERT'in genel sınıflandırma yeteneğinin daha iyi olduğunu

gösterir. Train Time: Eğitim süresi açısından ALBERT, DistilBERT'ten yaklaşık 2 kat daha hızlıdır. Bu, ALBERT'in daha verimli ve optimize edilmiş bir yapıya sahip olmasından kaynaklanmaktadır. Inference Time: Çıkarım süresi açısından DistilBERT daha hızlıdır (0.0349 saniye vs 0.0473 saniye). DistilBERT'in hafif yapısı burada avantaj sağlar. DistilBERT: Daha yüksek doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall) ve F1 skoruna sahiptir. Çıkarım süresi daha hızlıdır, bu da gerçek zamanlı uygulamalar için avantaj sağlar. Ancak, eğitim süresi ALBERT'e göre çok daha uzundur. ALBERT: Eğitim süresi açısından oldukça hızlıdır, bu da özellikle büyük veri setleri üzerinde çalışırken önemli bir avantaj sağlar. AUC metriğinde daha iyidir, bu da sınıflandırma doğruluğunun genel olarak yüksek olduğunu gösterir. Negatif sınıfları doğru tespit etmede (specificity) DistilBERT'ten daha başarılıdır. Eğer yüksek performans (Accuracy, Precision, Recall, F1 Score) ve hızlı çıkarım (Inference Time) önemliyse, DistilBERT tercih edilmelidir. Eğer eğitim süresi kısıtlı ve genel sınıflandırma başarısı (AUC) ön plandaysa, ALBERT daha uygun bir seçimdir. Özellikle büyük ölçekli veri setleriyle çalışırken ALBERT'in eğitim süresi avantajı önemli bir fark yaratabilir.

G. ALBERT-GPT

GPT, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skorunda ALBERT'ten daha iyi sonuçlar vermiştir. Bu, GPT'nin tahminlerinin ALBERT'e göre daha dengeli olduğunu göstermektedir.

Sensitivity: GPT, pozitif sınıfları tespit etme yeteneğinde ALBERT'ten daha üstündür. Specificity: Negatif sınıfları doğru tespit etmede ALBERT, GPT'den daha başarılıdır. AUC metriğinde GPT, ALBERT'e göre çok küçük bir üstünlük sağlamaktadır. Bu, genel sınıflandırma yeteneği açısından GPT'nin ALBERT'ten biraz daha iyi olduğunu gösterir.

ALBERT, eğitim süresi açısından GPT'den daha hızlıdır. Çıkarım Süresi: GPT, çıkarım süresinde ALBERT'ten daha hızlıdır. ALBERT: Eğitim süresi açısından oldukça hızlıdır, bu da özellikle büyük veri setlerinde çalışırken önemli bir avantaj sağlar. Negatif sınıfları doğru tespit etme başarısı (specificity) yüksektir. GPT: Daha yüksek doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoruna sahiptir. Pozitif sınıfları tespit etme (sensitivity) yeteneği daha yüksektir. Çıkarım süresi daha hızlıdır ve genel sınıflandırma başarısı (AUC) açısından ALBERT'ten biraz daha iyidir. ALBERT: Eğer eğitim süresi kısıtlıysa ve negatif sınıfları tespit etme başarısı (Specificity) önemliyse, ALBERT tercih edilmelidir. Ayrıca büyük ölçekli veri setlerinde daha verimli bir seçim olabilir. GPT: Eğer genel sınıflandırma başarısı (Accuracy, AUC) ve pozitif sınıfları tespit etme yeteneği (Sensitivity) ön plandaysa, GPT daha uygun bir seçimdir. Ayrıca çıkarım süresi daha hızlı olduğundan gerçek zamanlı uygulamalarda avantaj sağlayabilir.

H. ROBERTA-DISTILBERT

Performans olarak RoBERTa, doğruluk (accuracy), hassasiyet (precision), geri çağırma (recall) ve F1 skorunda Distil-

BERT'ten daha iyi sonuçlar vermiştir. DistilBERT daha hafif bir model olduğundan, RoBERTa kadar yüksek performans sağlayamasa da makul sonuçlar sunmaktadır. Sensitivity (Duyarlılık): DistilBERT, duyarlılık metriğinde RoBERTa'ya göre bir miktar daha başarılıdır. Bu, DistilBERT'in pozitif sınıfları biraz daha iyi tespit ettiğini gösterir. Specificity (Özgüllük): RoBERTa, negatif sınıfları doğru bir şekilde tespit etmede daha başarılıdır. RoBERTa, AUC metriğinde DistilBERT'ten daha üstündür. Bu, RoBERTa'nın genel sınıflandırma yeteneğinin daha iyi olduğunu gösterir. Train Time: RoBERTa, eğitim süresi açısından DistilBERT'ten yaklaşık 3 kat daha hızlıdır. Inference Time: Çıkarım süresi açısından DistilBERT çok daha hızlıdır. Bu, DistilBERT'in hafif yapısı sayesinde gerçekleşmektedir. Performans odaklı bir yaklaşım istiyorsa, RoBERTa daha iyi bir seçimdir. Genel doğruluk, hassasiyet, geri çağırma, F1 skoru ve AUC metriğinde DistilBERT'ten üstündür. Hız ve verimlilik odaklı bir yaklaşım arınıyorsa: Eğitim süresi açısından RoBERTa, DistilBERT'ten çok daha hızlıdır. Çıkarım süresi açısından ise DistilBERT, hafif yapısı sayesinde RoBERTa'ya göre %40 daha hızlı sonuçlar üretebilir. Öneri: Eğer yüksek doğruluk ve performans önemliyse, RoBERTa tercih edilmelidir. Eğer daha hızlı çıkarım süresi ve hafif bir model ihtiyaç varsa, DistilBERT iyi bir seçenektir. Özellikle donanım sınırlıysa (örneğin mobil cihazlarda) DistilBERT öne çıkar.

I. ROBERTA-GPT

RoBERTa, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skorunda GPT'den daha iyi sonuçlar vermiştir. Bu, RoBERTa'nın tahminlerinin GPT'ye göre daha tutarlı ve dengeli olduğunu göstermektedir. Sensitivity: Pozitif sınıfları tespit etme yeteneğinde GPT, RoBERTa'dan daha üstündür. Specificity: Negatif sınıfları doğru tespit etmede RoBERTa, GPT'den daha başarılıdır. AUC metriğinde GPT, RoBERTa'dan biraz daha iyi bir performans sergilemiştir. Bu, genel sınıflandırma yeteneği açısından GPT'nin az da olsa daha iyi olduğunu gösterir.

Eğitim Süresi: RoBERTa, eğitim süresi açısından GPT'den daha hızlıdır. Çıkarım Süresi: GPT, çıkarım süresinde RoBERTa'dan daha hızlıdır. RoBERTa: Daha yüksek doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoru ile genel performans açısından GPT'ye göre daha iyi bir seçenektir. Negatif sınıfları doğru tespit etme başarısı (specificity) daha yüksektir. Eğitim süresi daha kısadır, bu da büyük veri setlerinde avantaj sağlar. GPT: Pozitif sınıfları tespit etme yeteneği (sensitivity) daha yüksektir. Genel sınıflandırma başarısı (AUC) açısından çok az bir üstünlüğe sahiptir. Çıkarım süresi daha hızlıdır ve gerçek zamanlı uygulamalarda avantaj sağlar. RoBERTa: Eğer genel performans metrikleri (Accuracy, Precision, Recall, F1 Score) ve negatif sınıfları doğru tespit etme başarısı (Specificity) ön plandaysa, RoBERTa tercih edilmelidir. Ayrıca eğitim süresinin daha kısa olması büyük veri setleri için avantaj sağlayabilir. GPT: Eğer pozitif sınıfları tespit etme yeteneği (Sensitivity) ve hızlı çıkarım (Inference Time) önemliyse, GPT daha uygun bir seçimdir. Ayrıca

AUC metriğindeki üstünlük, genel sınıflandırma doğruluğu açısından GPT'yi öne çıkarabilir.

J. DISTILBERT-GPT

DistilBERT, doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoru metriklerinde GPT'den daha iyi sonuçlar vermiştir. Bu, DistilBERT'in tahminlerinin daha tutarlı ve dengeli olduğunu göstermektedir.

Sensitivity: Pozitif sınıfları tespit etme yeteneğinde GPT, DistilBERT'e göre daha üstündür. Specificity: Negatif sınıfları doğru tespit etmede DistilBERT, GPT'den daha başarılıdır. AUC metriğinde GPT, DistilBERT'e göre daha iyi bir performans sergilemiştir. Bu, genel sınıflandırma yeteneği açısından GPT'nin daha üstün olduğunu gösterir.

Eğitim Süresi: GPT, eğitim süresi açısından DistilBERT'ten neredeyse iki kat daha hızlıdır. Çıkarım Süresi: DistilBERT, çıkarım süresinde GPT'den daha hızlıdır. Bu da gerçek zamanlı uygulamalarda avantaj sağlayabilir. DistilBERT: Daha yüksek doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoru ile genel performans açısından GPT'ye göre daha iyi bir seçenektir. Negatif sınıfları doğru tespit etme başarısı (specificity) daha yüksektir. Çıkarım süresi daha hızlıdır, bu da gerçek zamanlı uygulamalarda avantaj sağlar. GPT: Pozitif sınıfları tespit etme yeteneği (sensitivity) açısından daha üstündür. Genel sınıflandırma başarısı (AUC) metriğinde üstünlük sağlamıştır. Eğitim süresi oldukça kısadır, bu da büyük veri setlerinde veya zaman kısıtlaması olan projelerde avantaj sunar. DistilBERT: Eğer genel performans metrikleri (Accuracy, Precision, Recall, F1 Score) ve negatif sınıfları doğru tespit etme başarısı (Specificity) ön plandaysa, DistilBERT tercih edilmelidir. Çıkarım süresinin daha hızlı olması da gerçek zamanlı uygulamalar için DistilBERT'i daha uygun hale getirir. GPT: Eğer pozitif sınıfları tespit etme yeteneği (Sensitivity), daha yüksek genel sınıflandırma başarısı (AUC) ve kısa eğitim süresi önemliyse, GPT daha uygun bir seçimdir. Özellikle sınırlı eğitim süresi gerektiren veya büyük veri setleriyle çalışılan durumlarda GPT avantajlıdır.

REFERENCES

- [1] Vaswani, A., et al. (2017). "Attention Is All You Need." Advances in Neural Information Processing Systems (NeurIPS).
- [2] Mitchell, R. (2018). Web Scraping with Python: Collecting Data from the Modern Web. O'Reilly Media.
- [3] Hugging Face Transformers Library, <https://huggingface.co/docs/transformers/index>
- [4] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [5] Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-Art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
- [6] Radford, A., Wu, J., Child, R., et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI. <https://cdn.openai.com/better-language-models/>