

# Kocaeli Üniversitesi

## Bilişim Sistemleri Mühendisliği

### Makine Öğrenmesine Giriş Dersi

#### DOĞAL DİL İŞLEME İLE ŞİİRLERDEN UYGUN AKIMI TESPİT ETME PROJE RAPORU

Kaan TORAMAN  
211307078  
Teknoloji Fakültesi  
KOCAELİ Üniversitesi  
İzmit, Kocaeli/ Türkiye  
[kaan41tor@gmail.com](mailto:kaan41tor@gmail.com)

**Abstract**—This paper details the process of collecting poems from Turkish history through the Selenium library and then teaching them to natural language processing (NLP) models. This process was carried out using five different Transformer language processing models. The focus of the report is on the methods used in the collection, processing and modeling of the poems, and the characteristics and performance of the Transformer models used.

**Keywords**—SELENIUM, NLP, POEM, Transformers

**Özet**—Bu rapor, Türk tarihinde yer edinmiş şiirlerin, Selenium kütüphanesi aracılığıyla toplanması ve ardından doğal dil işleme modellerine öğretilmesi sürecini detaylı bir şekilde ele almaktadır. Bu süreç, beş farklı Transformer dil işleme modeli kullanılarak gerçekleştirilmiştir. Şiirlerin derlenmesi, işlenmesi ve modellenmesi süreçlerinde hangi yöntemlerin tercih edildiği, kullanılan Transformer modellerin özellikleri ve performansları, raporun odak noktalarından birini oluşturmaktadır.

**Keywords**—SELENIUM, NLP, Şiir, Transformers

#### 1. AMAÇLAR VE KAPSAM

Bu proje, Türk şiirlerinin tarihsel ve kültürel zenginliğini doğal dil işleme (NLP) modelleri aracılığıyla incelemeyi ve şiirlerden uygun edebi akımı tespit etmeyi hedeflemektedir. Selenium kütüphanesi kullanılarak çeşitli kaynaklardan toplanan şiirler, beş farklı Transformer modeli olan RoBERTa, DistilBERT, BERT, Electra ve OpenAI GPT- 2 derin öğrenme modellerine öğretilmiştir. Projenin amaçları doğrultusunda aşağıdaki sorulara cevaplar bulunması hedeflenmiştir:

- Türk şiirlerinin hangi edebi akımlara ait olduğu NLP modelleriyle belirlenebilir mi?
- Şiirlerin içerdiği temalar ve dil özellikleri Transformer modelleri tarafından ne kadar başarılı bir şekilde sınıflandırılmaktadır?
- Her bir Transformer modelinin Türk şiirlerini analiz etme ve uygun akımı tespit etme performansı nasıldır?
- Türk şiirlerinin modern NLP teknikleriyle incelenmesi, edebiyat araştırmalarına ve edebi akım analizlerine nasıl katkı sağlayabilir?

#### 2. VERİ TOPLAMA

Proje, Türk şiirlerinin analiz edilmesine odaklanarak, bağımsız olduğu düşünülen "Şiir Sitesi" [1] adresi üzerinden toplanan şiirlerle gerçekleştirilmiştir. Bu site, geniş bir Türk şiiri koleksiyonunu barındırmakta olup, projede kullanılacak verilerin çeşitliliği ve kalitesi açısından önemli bir kaynak teşkil etmektedir. Burada verilen kaynakta bulunan, bütün şairlerin şiirleri toplanmıştır ve .csv uzantılı olarak kaydedilmiştir.

Verilerin toplandığı kaynaktan şiirlerin alınması için Python dilinde kod yazılmıştır ve JAVA dilinde geliştirilen "SELENIUM" [2] kütüphanesi kullanılmıştır. Bu işlemler için kaynak sayfanın tasarımında kullanılan HTML kodlarından yararlanılmıştır. Sayfa iki sütuna bölünmüş, her sütun içindeki yazar başlıkları bir diziyeye atanmış ve yazarlara ait eserlerin bulunduğu sayfaya erişilmiştir. Ardından eserler sayfası içinde aynı işlemler tekrarlanıp şiir verileri teker teker indirilmiştir. İndirilen veriler ".csv" uzantısıyla, yazar adı, şiir başlığı ve şiir sıralamasında kaydedilmektedir.

##### 2.1. SONUÇLARIN TEMİZLENMESİ

Python dilinde yazılan yazılım sonucunda oluşturulan .csv dosyası, tekrardan Python yazılım diliyle oluşturulmuş bir algoritma ile temizlenmiştir. Oluşturulan .csv algoritma tarafından açılır. Açılan algoritmada sırasıyla şu işlemler gerçekleştirir:

- Bütün metin verilerinin başlarında ve sonlarında boşluk varsa kaldırılır.
- Bütün karakterler, küçük karakterlerle değiştirilir.
- Bütün metin verilerinden yabancı karakterler arındırılır.
- Bütün metin verilerinden, emoji ve noktalama işaretleri arındırılır.

Not: Burada bahsedilen "metin verileri" kavramı içerisine yazar isimleri, şiir başlıkları ve şiirler dahildir.

Tüm bu işlemlerin ardından verilerin temizlenmiş halleri .csv olarak tekrardan kaydedilir.

## 2.2. VERİLERİN SINIFLANDIRILMASI

Verilerin sınıflandırılması kısmı, projenin sıkıntılı olan kısımlarından birisidir. Toplanmış olan verilerde, belirgin bir farklılık bulunamadığı için verilerin sınıflandırılması elle yapılmıştır. Şairlerin yazdıkları eserler bulundukları döneme göre etiketlenmiştir. Bu süreç içerisinde etiketlenmesi yapılamayan veriler bulunmuştur, bu veriler kullanılmamıştır.

## 2.3. VERİLERİN ÇOĞALTILMASI

Verilerin çoğaltılmasının ardından sınıflarda bulunan veri sayıları netliğe kavuşmuştur. Bu netlik sonucunda bir çok sınıfta, verilerin yetersiz olduğuna karar verilmiştir ve sınıf sayısının dörde düşürülme kararı alınmıştır. Sınıflarda bulunan verilerin sayısının eşit olması için sınıflarda çoğaltmalar ve azaltmalar yapılmıştır. Azaltmak için veriler silinmiştir. Çoğaltmak için ise Python dili ile bir algoritma yazılmıştır. Yazılan algoritma şu şekilde çalışmaktadır:

- Etiketlenmiş veriler bir döngü içerisinde sırayla çağırılır.
- Döngü içerisinde çeviri işlemleri yapılır.
- Orijinali Türkçe olan eserler sırasıyla İngilizce-Türkçe, Fransızca-Türkçe, İzlandaca-Türkçe, İspanyolca-Türkçe ve Almanca-Türkçe dillerinde çevrilir.
- Çevirilir sonucunda elde edilen metin verileri tekrardan orijinal diline çevrilir.
- Orijinal diline çevirilen metin verileri, önceden elimizde bulunan .csv veri seti üzerine eklenir.

Tüm bu işlemlerin ardından verilerin çoğaltmış halleri .csv olarak tekrardan kaydedilir.

## 2.4. SONUÇ

Veri toplama sonucunda **5461** şiir indirilmiştir. Bu veriler yetersiz bulunarak **3085** veri oluşturulmuştur. Eğitim süreci için her sınıfta **1000** veri olacak şekilde **4** sınıf kullanılmıştır. Diğer istatistiksel bilgiler aşağıda sıralanmıştır:

Toplanan Veriler

Sınıf	Veri Sayısı
Beş Hececiler	50
Garip Akımı	536
İkinci Yeni	462
Milli Edebiyat Dönemi	22
Modern Türk	19
Saf Şiir	197
Servet-i Fünun	7
Toplumcu Gerçekçi	896
Yedi Meşaleciler	43
Etiketlenmemiş	3229

Üretilen Veriler

Dil	Veri Sayısı
İngilizce	1926
Fransızca	598
Almanca	189
İzlandaca	186
İspanyolca	190

## 3. PROJENİN MODELLENMESİ

Projenin modellenmesi sürecinde beş farklı Transformer modeli ile çalışılmıştır. Bu modeller sırasıyla BERT, DistilBERT, RoBERTa, Electra ve OpenAI GPT Transformer modelleridir.

Proje boyunca kullanılan veri seti sıkıntılı bir yapıya sahip olduğu için ezberleme yapmaya müsaittir. Bu nedenle aşırı öğrenme (overfitting) yapmasını engellemek için çeşitli yöntemler uygulanmıştır. Kod içerisinde **k-fold cross validation** kullanılmıştır. Bu yöntemde, veri seti k adet daha küçük alt sete bölünür ve her alt set sırasıyla bir doğrulama seti olarak kullanılırken geri kalan k-1 alt set eğitim için kullanılır. Bu işlem k kez tekrarlanır ve her seferinde farklı bir alt set doğrulama seti olarak seçilir. Kodlama aşamasında k değeri **5** olarak tercih edilmiştir. Bu sayede her fold için **%80** eğitim ve **%20** deneme verisi elde edilmektedir. **Erken durdurma (Early Stopping)** yöntemi, modelin belirli bir sayıda epoch boyunca iyileşme göstermediğinde eğitimi durdurarak gereğinden fazla eğitilmesini ve dolayısıyla aşırı öğrenme yapmasını önler. Bu çalışmada erken durdurma parametresi **3** olarak belirlenmiştir. Ayrıca, Dropout yöntemi ile her eğitim adımında modelin bazı bağlantıları rastgele devre dışı bırakılmış, bu da modelin genelleme yeteneğini artırmıştır. **Ağırlık azaltma (Weight Decay)** yöntemi ise modelin ağırlıklarının büyümesini sınırlayarak daha basit modellerin tercih edilmesine ve aşırı öğrenmenin azaltılmasına yardımcı olmuştur. **Hiperparametre optimizasyonu (Grid Search)** ile farklı hiperparametre kombinasyonları deneyerek en iyi performans gösteren kombinasyon bulunmuş ve böylece modelin performansı artırılarak aşırı öğrenme riski azaltılmıştır. Bu yöntemler, aracılığıyla modellerin genelleme yeteneğini artırılarak daha doğru ve güvenilir sonuçlar elde edilmesi hedeflenmiştir.

Modeller için yazılmış olan kodlar içerisinde çeşitli kütüphaneler bulunmaktadır. Bu kütüphaneler:

- **numpy**: Sayısal hesaplamalar için kullanılır.
- **pandas**: Veri manipülasyonu ve analizi için kullanılır.
- **matplotlib**: Grafik ve görselleştirme için kullanılır.
- **sklearn**: Makine öğrenimi algoritmaları ve araçları sağlar. Burada özellikle veri bölme ve performans değerlendirme metrikleri için kullanılmıştır.
- **torch**: PyTorch kütüphanesi, derin öğrenme modelleri oluşturmak ve eğitmek için kullanılır.
- **transformers**: Hugging Face kütüphanesi, BERT, DistilBERT, RoBERTa, Electra ve GPT gibi önceden eğitilmiş modelleri kullanarak metin

sınıflandırması ve diğer NLP görevlerini gerçekleştirmek için kullanılır.

Kodumuzda belirtilen kütüphaneler içeri aktarıldıktan sonra, sırasıyla şu işlemler gerçekleştirilir: veri seti yükleme ve hazırlama, etiketlerin sayısal değerlere dönüştürülmesi, modelin yüklenmesi, veri seti için özel bir sınıf tanımlanması, performans metriklerini hesaplamak için fonksiyon oluşturulması, sonuçların görselleştirilmesi ve grid search yardımıyla eğitim fonksiyonunun çağırılması.

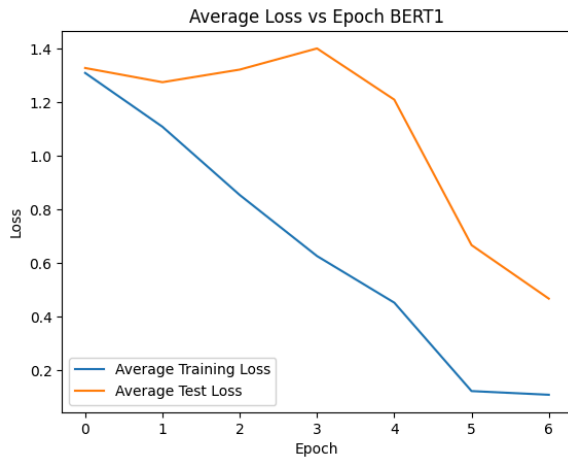
### 3.1. BERT Transformer Modeli

BERT (Bidirectional Encoder Representations from Transformers) [3], Google tarafından geliştirilen ve doğal dil işleme (NLP) performans gösteren bir derin öğrenme modelidir. BERT, kelimelerin anlamını hem öncesinde bulunun hem de sonrasında bulunan kelimelerle birlikte değerlendirir. Bu sayede bağlamsal anlamı daha doğru bir şekilde belirler. Büyük metin veri setlerinde ön eğitim ile öğretilen BERT, ince ayarlar yapılarak çeşitli NLP işlemlerinde kullanılabilir. BERT'in Transformer mimarisinin encoder olarak kullanılması, uzun bağımlılıkları ve ilişkileri modellemede başarılı olmasını sağlar.

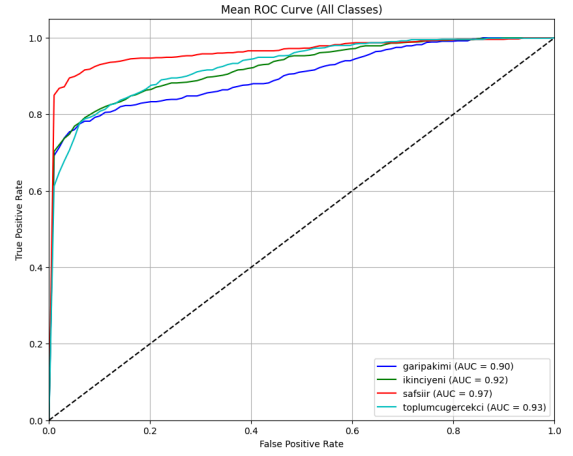
BERT modelinde, kullanılan parametreler aşağıdaki gibidir:

Epoch	10
Max Length	256
Fold	5
Early Stopping Patience	3
Learning Rate	2e-5
Weight Decay	0.01
Dropout Prob	0.1
Warmup Steps	500
Batch Size	8

BERT modeline ait ROC Curve ve Loss vs Epoch grafikleri aşağıdaki gibidir:



Şekil 1 - BERT Loss vs Epoch grafiği 1



Şekil 2 - BERT ROC Curve Grafiği

BERT modelinin 5 fold için çıktıları:

Fold	Accuracy	F1	Precision	Recall	Sensitivity	AUC
1	0,4875	0,491759	0,572493	0,4875	0,4875	0,9
2	0,4825	0,470736	0,552379	0,4825	0,4825	0,92
3	0,505	0,500474	0,582493	0,505	0,505	0,97
4	0,58375	0,574329	0,595225	0,58375	0,58375	0,93
5	0,49	0,47602	0,508923	0,49	0,49	0,84
Avg	0,451999	0,446385	0,464674	0,451999	0,451999	0,92

Şekil 3 - BERT Çıktıları

BERT Modelinde:

- **Ortalama Epoch Süresi:** 420 saniye (7 dakika)
- **Ortalama Fold Süresi:** 2520 saniye (42 dakika)
- **Toplam Çalışma Süresi:** 12600 saniye (210 dakika)

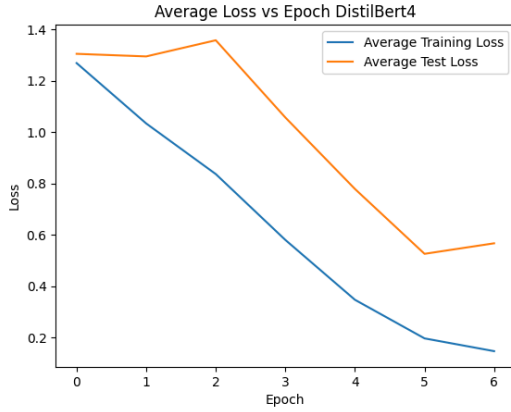
### 3.2. DistilBERT Transformer Modeli

DistilBERT (Distilled BERT) [4], BERT transformer modelinin daha hafif ve hızlı versiyonudur. Hugging Face tarafından geliştirilmiştir. DistilBERT, BERT modelinin bilgi yoğunluğunu korurken, daha az parametre ile daha hızlı ve verimli çalışmayı hedefler. DistilBERT, kelimelerin anlamını hem öncesinde bulunun hem de sonrasında bulunan kelimelerle birlikte değerlendirir. Bu sayede bağlamsal anlamı daha doğru bir şekilde belirler. Büyük metin veri setlerinde ön eğitim ile öğretilen DistilBERT, ince ayarlar yapılarak çeşitli NLP işlemlerinde kullanılabilir. DistilBERT, BERT orijinal modeline kıyasla %40 daha küçük ve %60 daha hızlıdır, ancak performans açısından benzer sonuçlar sunar.

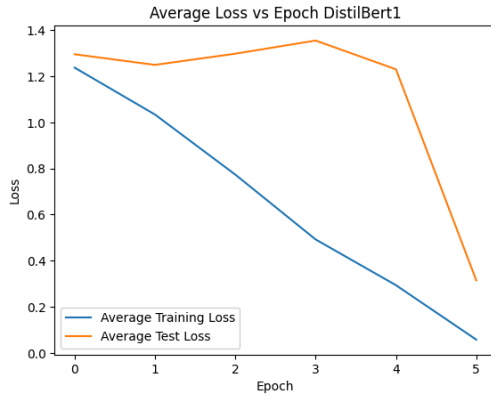
DistilBERT modelinde, kullanılan parametreler aşağıdaki gibidir:

Epoch	10
Max Length	256
Fold	5
Early Stopping Patience	3
Learning Rate	2e-5
Weight Decay	0.0
Dropout Prob	0.1
Warmup Steps	0
Batch Size	16

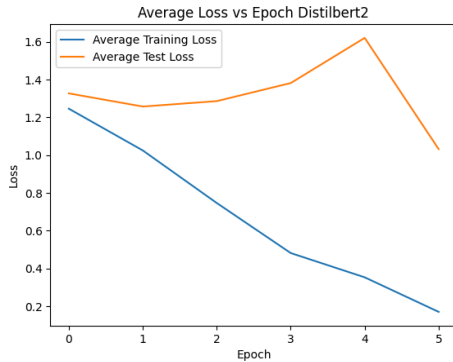
DistilBERT modeline ait ROC Curve ve Loss vs Epoch grafikleri aşağıdaki gibidir:



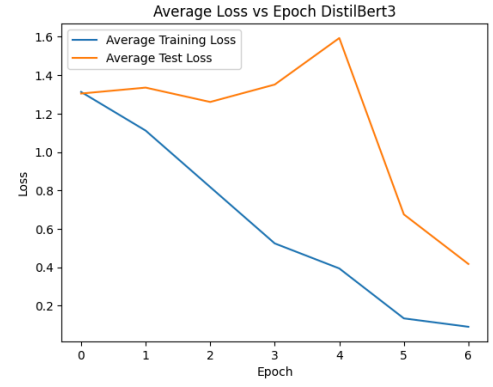
Şekil 4 - DistilBERT Loss vs Epoch grafiği 1



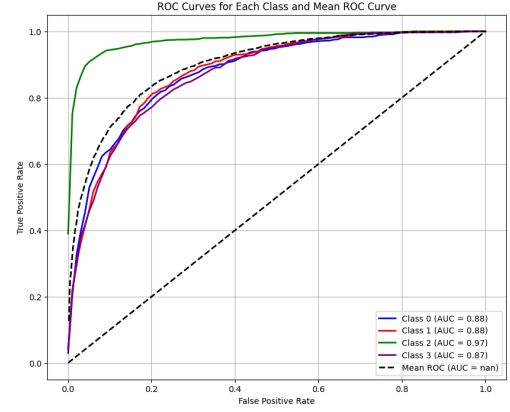
Şekil 5 - DistilBERT Loss vs Epoch grafiği 2



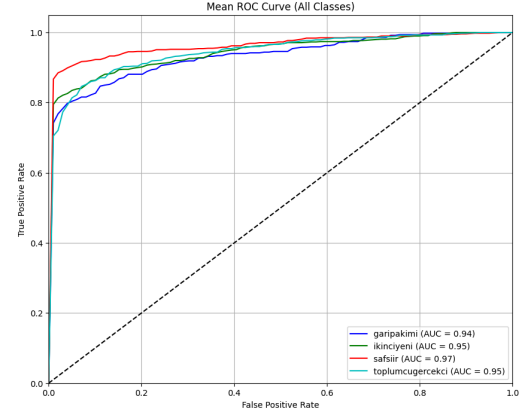
Şekil 6 - DistilBERT Loss vs Epoch grafiği 3



Şekil 7 - DistilBERT Loss vs Epoch grafiği 4



Şekil 8 - DistilBERT ROC Curve Grafiği 1



Şekil 9 - DistilBERT ROC Curve Grafiği 2

DistilBERT modelinin 5 fold için çıktıları:

Fold	Accuracy	F1	Precision	Recall	Sensitivity	AUC
1	0,395	0,388335	0,449414	0,395	0,395	0,86
2	0,47125	0,470728	0,496665	0,47125	0,47125	0,87
3	0,54125	0,542542	0,55055	0,54125	0,54125	0,97
4	0,58	0,565595	0,635711	0,58	0,58	0,87
5	0,46	0,463621	0,474016	0,46	0,46	0,87
Avg	0,4865	0,472063	0,495787	0,4865	0,4865	0,8925

Şekil 10 - DistilBERT Çıktıları

DistilBERT modelinde:

- **Ortalama Epoch Süresi:**100.32 saniye(1.7 dakika)
- **Ortalama Fold Süresi:** 1003.2 saniye (17 dakika)
- **Toplam Çalışma Süresi:** 5016 saniye (84 dakika)

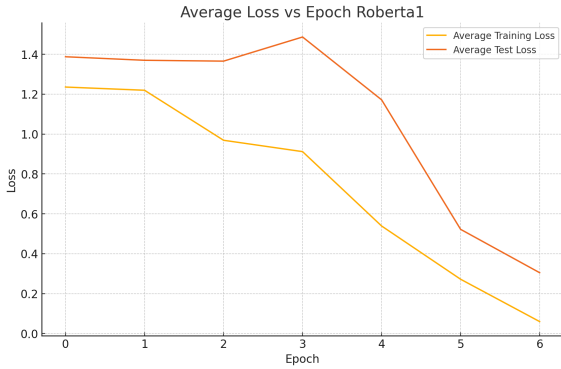
### 3.3. RoBERTa Transformer Modeli

RoBERTa (Robustly optimized BERT approach) [5], BERT modelinin daha güçlü ve optimize edilmiş versiyonudur. FacebookAI tarafından geliştirilmiştir. RoBERTa, BERT'in temel mimarisini kullanır ancak eğitim verisi miktarını artırarak, daha uzun eğitim süresi ve farklı hiperparametrelerle performansı iyileştirir. RoBERTa, kelimelerin anlamını hem öncesinde bulunun hem de sonrasında bulunan kelimelerle birlikte değerlendirir. Bu sayede bağlamsal anlamı daha doğru bir şekilde belirler. Büyük metin veri setlerinde ön eğitim ile öğretilen RoBERTa, ince ayarlar yapılarak çeşitli NLP işlemlerinde kullanılabilir. RoBERTa, veri ön işleme ve eğitim prosedürlerinde yapılan iyileştirmeler sayesinde, birçok NLP görevinde BERT'ten daha iyi performans gösterir.

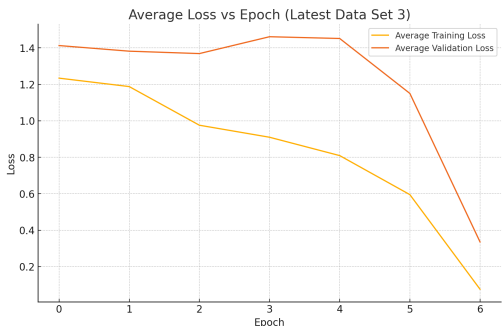
RoBERTa modelinde, kullanılan parametreler aşağıdaki gibidir:

Epoch	10
Max Length	256
Fold	5
Early Stopping Patience	3
Learning Rate	2e-5
Weight Decay	0.0
Dropout Prob	0.1
Warmup Steps	500
Batch Size	8

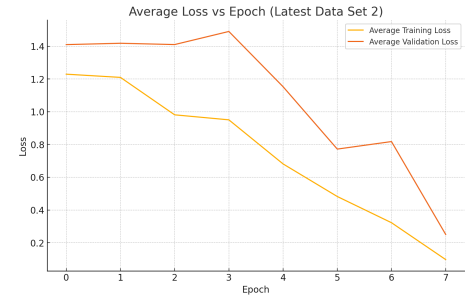
RoBERTa modeline ait ROC Curve ve Loss vs Epoch grafikleri aşağıdaki gibidir:



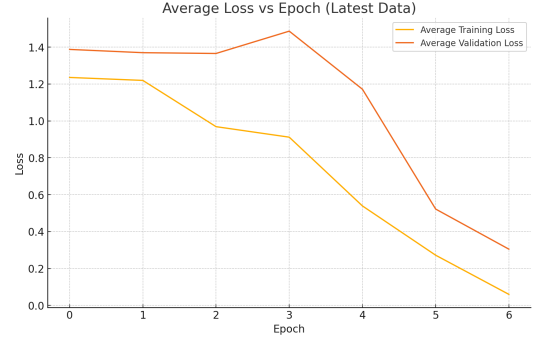
Şekil 11- RoBERTa Loss vs Epoch grafiği 1



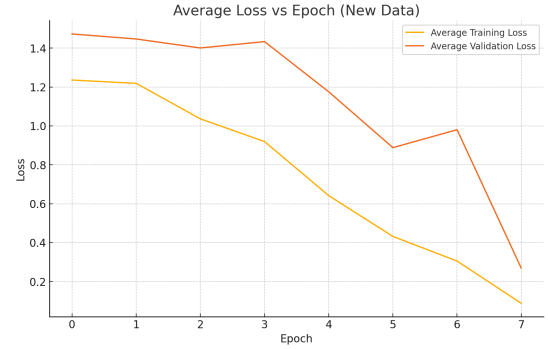
Şekil 12 - RoBERTa Loss vs Epoch grafiği 2



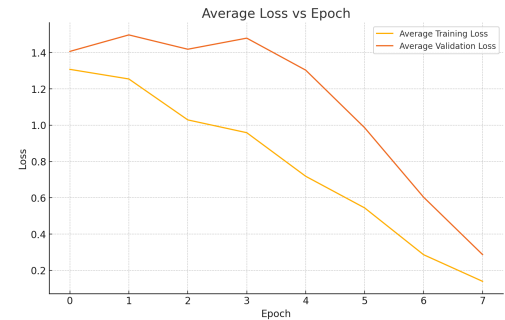
Şekil 13 - RoBERTa Loss vs Epoch grafiği 3



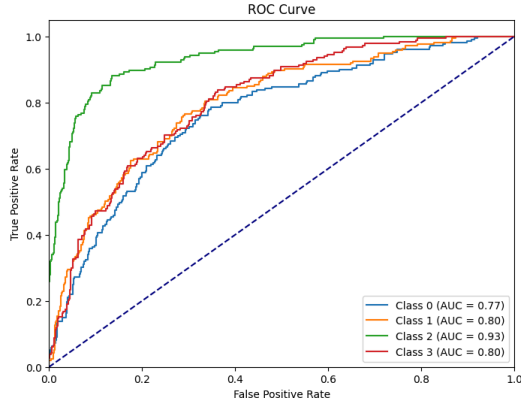
Şekil 14 - RoBERTa Loss vs Epoch grafiği 4



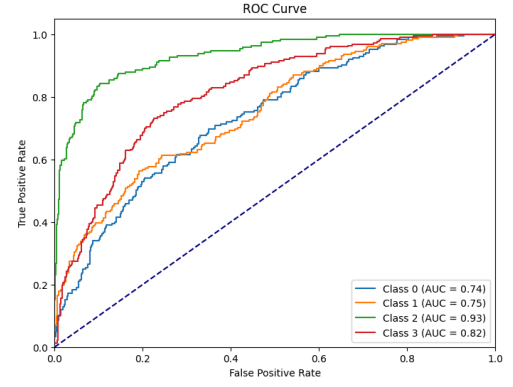
Şekil 15 - RoBERTa Loss vs Epoch grafiği 5



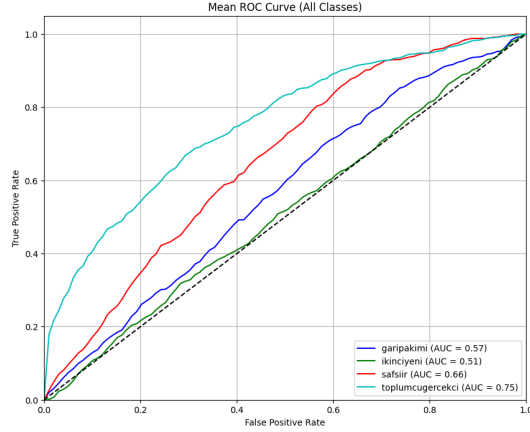
Şekil 16 - RoBERTa Loss vs Epoch grafiği 6



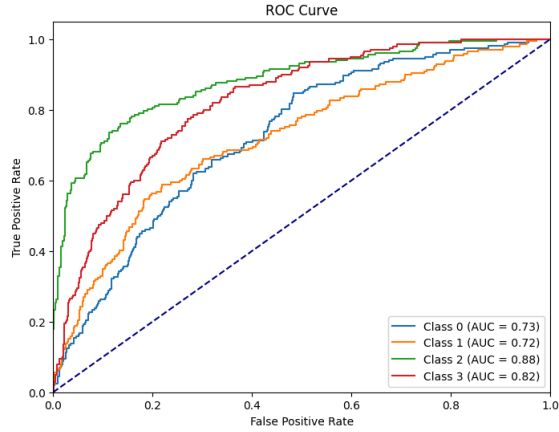
Şekil 17 - RoBERTa ROC Curve Grafiği 1



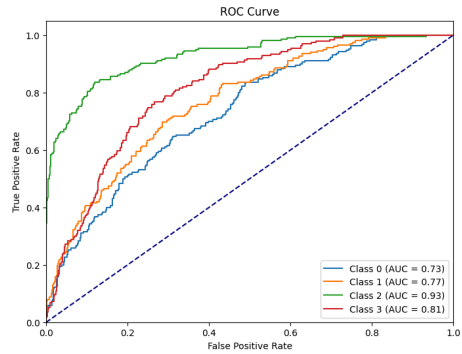
Şekil 21 - RoBERTa ROC Curve Grafiği 5



Şekil 18 - RoBERTa ROC Curve Grafiği 2



Şekil 19 - RoBERTa ROC Curve Grafiği 3



Şekil 20 - RoBERTa ROC Curve Grafiği 4

RoBERTa modelinin 5 fold için çıktıları:

Fold	Accuracy	F1	Precision	Recall	Sensitivity	AUC
1	0,3425	0,307065	0,425558	0,3425	0,3425	0,86
2	0,2525	0,174696	0,257737	0,2525	0,2525	0,87
3	0,5775	0,579635	0,605667	0,5775	0,5775	0,97
4	0,58625	0,59165	0,646283	0,58625	0,58625	0,87
5	0,35125	0,306098	0,358358	0,35125	0,35125	0,87
Avg	0,4445	0,416287	0,467999	0,4445	0,4445	0,8925

Şekil 22- RoBERTa Çıktıları

RoBERT Modelinde:

- **Ortalama Epoch Süresi:** 93.58 saniye (1.56 dakika)
- **Ortalama Fold Süresi:** 935.8 saniye ( 15.6 dakika)
- **Toplam Çalışma Süresi:** 4679 saniye (78 dakika)

### 3.4. Electra Transformer Modeli

Electra (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) [6], Google tarafından geliştirilen bir transformer modelidir. Electra, dil modelleme ve metin anlama görevlerinde yüksek performans sağlar. Electra, geleneksel maskeli dil modellemeden farklı olarak, belirli kelimeleri değiştirir ve bu kelimelerin doğru olup olmadığını tahmin eder. İki bileşenden oluşur: maskelenmiş kelimeleri tahmin eden “generator” ve bu tahminlerin doğru olup olmadığını belirleyen “discriminator”. Bu yöntem, daha az veri ve hesaplama gücü ile yüksek doğruluk sağlar.

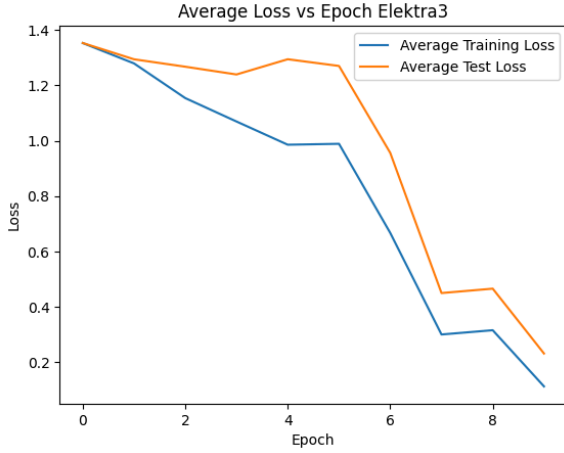
Electra modelinde, kullanılan parametreler aşağıdaki gibidir:

Epoch	10
Max Length	256
Fold	5
Early Stopping Patience	3
Learning Rate	2e-5
Weight Decay	0.0
Dropout Prob	0.1
Warmup Steps	500

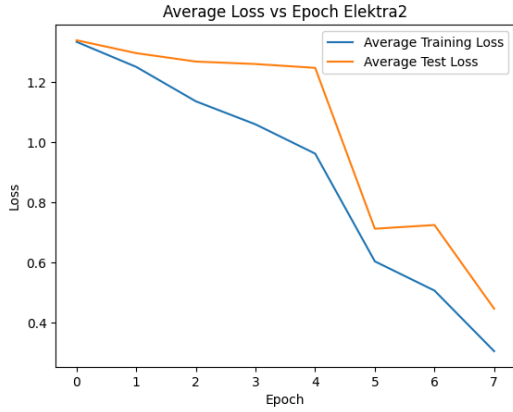
Batch Size

8

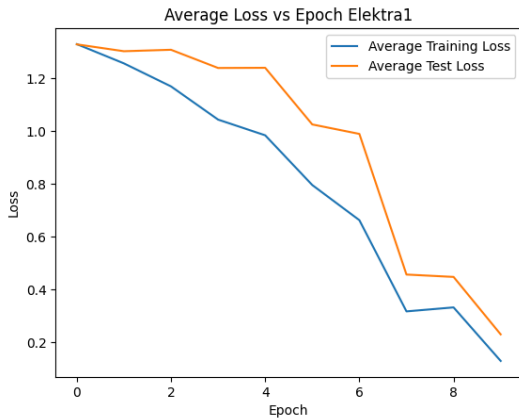
Electra modeline ait ROC Curve ve Loss vs Epoch grafikleri aşağıdaki gibidir:



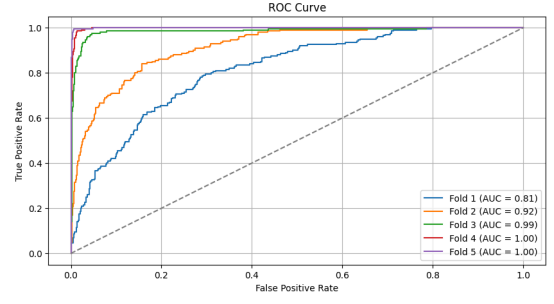
Şekil 23 - Elektra Loss vs Epoch grafiği 1



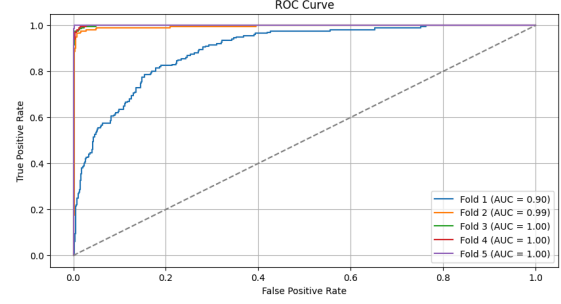
Şekil 24 - Elektra Loss vs Epoch grafiği 2



Şekil 25 - Elektra Loss vs Epoch grafiği 3



Şekil 26 - Elektra ROC Curve grafiği 1



Şekil 27 - Elektra ROC Curve grafiği

Electra modelinin 5 fold için çıktıları:

Fold	Accuracy	F1	Precision	Recall	Sensitivity	AUC
1	0.48375	0.487497	0.544826	0.48375	0.48375	0.9
2	0.45875	0.438178	0.46491	0.45875	0.45875	0.92
3	0.5225	0.520311	0.541847	0.5225	0.5225	0.97
4	0.5	0.486985	0.524588	0.5	0.5	0.93
5	0.41875	0.416217	0.445821	0.41875	0.41875	0.84
Avg	0.4865	0.472063	0.495787	0.4865	0.4865	0.92

Şekil 28 - Electra Çıktıları

Electra Modelinde:

- **Ortalama Epoch Süresi:** 300 saniye (5 dakika)
- **Ortalama Fold Süresi:** 3000 saniye (50 dakika)
- **Toplam Çalışma Süresi:** 15000 saniye (250 dakika)

### 3.5. OpenAI GPT-2 Transformer Modeli

GPT (Generative Pre-trained Transformer), OpenAI tarafından geliştirilen güçlü bir dil modelidir. GPT, büyük miktarda metin verisi üzerinde eğitilerek, insan benzeri metin üretme, dil anlama ve çeşitli dil görevlerini yerine getirme yeteneklerine sahiptir. Transformer mimarisini kullanarak, metin içindeki uzun vadeli bağımlılıkları ve ilişkileri öğrenir. GPT, "self-attention" mekanizması sayesinde metin bağlamını dikkate alarak yüksek kaliteli ve tutarlı metinler oluşturabilir. Bu model, önceden eğitilmiş olup, belirli bir görev için ince ayar yapılarak geniş bir uygulama yelpazesinde kullanılabilir.

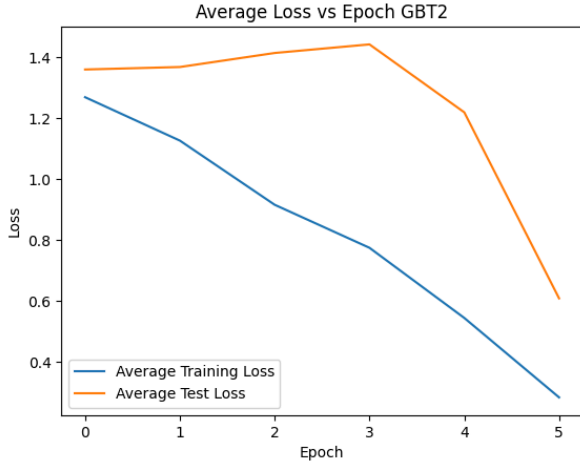
GBT modelinde, kullanılan parametreler aşağıdaki gibidir:

Epoch	10
Max Length	256
Fold	5
Early Stopping Patience	3

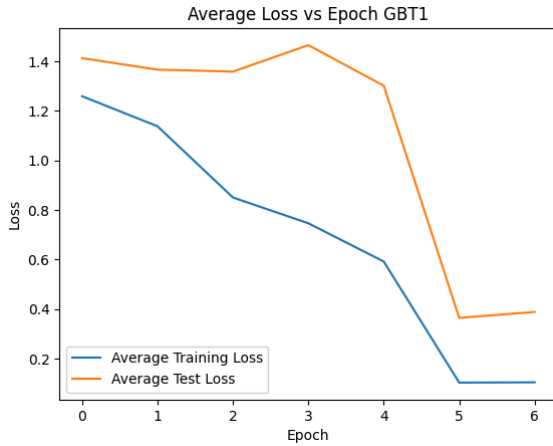


Learning Rate	2e-5
Weight Decay	0.01
Dropout Prob	0.1
Warmup Steps	500
Batch Size	8

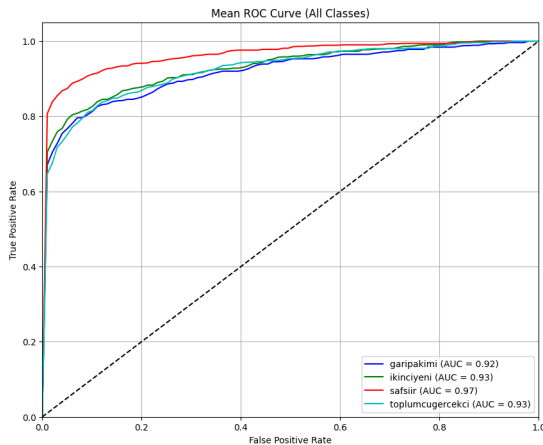
GBT modeline ait ROC Curve ve Loss vs Epoch grafikleri aşağıdaki gibidir:



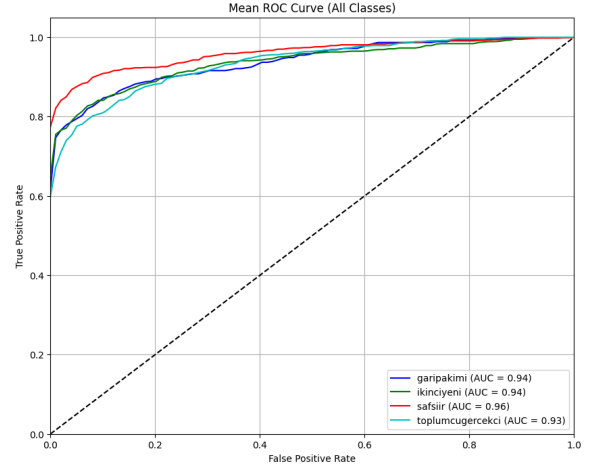
Şekil 29 - GBT Loss vs Epoch grafiği 1



Şekil 30 - GBT Loss vs Epoch grafiği 1



Şekil 31 - GBT ROC Curve Grafiği 1



Şekil 32 - GBT ROC Curve Grafiği 2

GBT modelinin 5 fold için çıktıları:

Fold	Accuracy	F1	Precision	Recall	Sensitivity	AUC
1	0,3475	0,290922	0,432118	0,3475	0,3475	0,86
2	0,39	0,36708	0,417948	0,39	0,39	0,87
3	0,5175	0,515474	0,552325	0,5175	0,5175	0,97
4	0,5425	0,545951	0,563321	0,5425	0,5425	0,87
5	0,40125	0,383031	0,487799	0,40125	0,40125	0,87
Avg	0,438	0,426771	0,467282	0,438	0,438	0,8925

Şekil 33 - GBT Çıktıları

GBT Modelinde:

- **Ortalama Epoch Süresi:** 105.88 saniye (yaklaşık 1.76 dakika)
- **Ortalama Fold Süresi:** 1058.8 saniye (yaklaşık 17.65 dakika)
- **Toplam Çalışma Süresi:** 5294 saniye (yaklaşık 88.23 dakika veya yaklaşık 1 saat 28.23 dakika)

#### 4. SONUÇLAR

Projenin sonuçlarına göre, RoBERTa modeli genel olarak en yüksek performansı göstermiştir. Bu model, eğitim verisi miktarını artırarak ve hiperparametre optimizasyonu yaparak diğer modellerden daha iyi sonuçlar elde etmiştir. DistilBERT modeli ise hız ve verimlilik açısından öne çıkmış ve BERT'de benzer performans göstermiştir. Electra modeli yüksek doğruluk sağlamış, ancak diğer modellerden daha uzun sürede çalışmıştır. OpenAI GPT-2 modeli ise dil anlama ve üretmede güçlü bir performans göstermiştir.

Sonuç olarak, Türk şiirlerinin edebi akımlarını tespit etmede RoBERTa modelinin en başarılı model olduğu; hız ve verimlilik açısından DistilBERT modelinin tercih edilebileceği; Electra ve GPT-2 modellerinin ise özel durumlarda kullanılabileceği belirlenmiştir.

NOT: RoBERTa modeli, diğer modellere kıyasla daha hızlı çalıştığı için bu model üzerinde daha fazla çalışma gerçekleştirilmiştir.

#### KAYNAKLAR



- [1] <https://siir.sitesi.web.tr/> [01.06.2024]
- [2] <https://www.selenium.dev/> [01.06.2024]
- [3] [https://huggingface.co/docs/transformers/model\\_doc/bert](https://huggingface.co/docs/transformers/model_doc/bert) [01.06.2024]
- [4] [https://huggingface.co/docs/transformers/model\\_doc/distilbert](https://huggingface.co/docs/transformers/model_doc/distilbert) [01.06.2024]
- [5] [https://huggingface.co/docs/transformers/en/model\\_doc/roberta](https://huggingface.co/docs/transformers/en/model_doc/roberta) [01.06.2024]
- [6] [https://huggingface.co/docs/transformers/en/model\\_doc/electra](https://huggingface.co/docs/transformers/en/model_doc/electra) [01.06.2024]
- [7] [https://huggingface.co/docs/transformers/model\\_doc/openai-gpt](https://huggingface.co/docs/transformers/model_doc/openai-gpt) [01.06.2024]