Kitap Türü Sınıflandırma: Metin Sınıflandırma Modelleri ve Vektörleştirme Yöntemlerinin Karşılaştırmalı Analizi

GİRİŞ

Bu proje, kitap açıklamalarından kitap türlerini tahmin etmeyi amaçlayan bir metin sınıflandırma problemini çözmek için farklı doğal dil işleme (NLP) tekniklerini kullanmaktadır. Modelimiz, metin verilerinin çeşitli vektörleştirme yöntemleri ile sayısal hale getirilmesini ve ardından bu sayısal verilerle bir sınıflandırma yapılmasını içermektedir. Amacımız, farklı vektörleştirme teknikleri ve derin öğrenme modelleri kullanarak en iyi doğruluk ve genelleme yeteneğine sahip bir model geliştirmektir.

Projede kullanılan temel bileşenler:

1. Veri Seti: Veri seti, kitapların açıklamalarından oluşmaktadır ve her açıklama, belirli bir kitap türüyle ilişkilidir.
2. Vektörleştirme Yöntemleri:
    - Bag of Words (BoW): Kelimelerin sıklıklarına dayalı olarak metni sayısal bir vektöre dönüştüren klasik bir yöntemdir.
    - TF-IDF (Term Frequency - Inverse Document Frequency): Kelime sıklığına ek olarak, kelimelerin tüm metin koleksiyonu içerisindeki önemini de dikkate alır.
    - Word2Vec: Kelimeleri sürekli değerli vektörlere dönüştüren bir yöntemdir. Bu yöntemde, kelimeler arasındaki anlamsal benzerlikler öğrenilir.
    - BERT (Bidirectional Encoder Representations from Transformers): Derin dil modelleri ailesinin bir üyesi olan BERT, kelimeler arasındaki bağlamı dikkate alarak metni vektörlere dönüştürür.
3. Modeller:
    - MLP (Multi-Layer Perceptron): Katmanlı bir yapıya sahip, derin öğrenme tabanlı bir yapay sinir ağı modelidir. BoW, TF-IDF, Word2Vec ve BERT vektörleriyle kullanılacak şekilde eğitilmiştir.
    - LSTM (Long Short-Term Memory): Zaman serisi verisi veya sıralı veriler üzerinde güçlü bir performans sergileyen bir tür RNN (Recurrent Neural Network) modelidir. Bu model, kelime sırasına dayalı verilerde daha başarılı sonuçlar vermektedir ve BoW, TF-IDF, Word2Vec ve BERT vektörleri ile uygulanmıştır.

Amaç

Bu projenin amacı, kitap açıklamalarını kullanarak en doğru ve verimli sınıflandırma modelini bulmaktır. Farklı vektörleştirme yöntemlerinin ve derin öğrenme modellerinin karşılaştırılmasıyla, hangi yaklaşımın daha iyi performans gösterdiği belirlenmeye çalışılacaktır.

Veri Ön İşleme

Veri Setinin Yüklenmesi ve Temizlenmesi

Projemiz, kitap türlerini tahmin etmek amacıyla kitap açıklamalarından oluşan bir veri setini kullanmaktadır. Veri setindeki her bir örnek, bir kitap açıklaması ve bu açıklamaya karşılık gelen kitap türü etiketinden oluşmaktadır. İlk adım, bu verilerin düzgün bir şekilde işlenmesi ve modelimize uygun hale getirilmesidir. Bu işlem aşağıdaki adımlardan oluşmaktadır:

1. **Veri Yükleme:** Veri seti, metin dosyalarından (TXT formatında) oluşmakta olup, her dosyada bir kitap açıklaması bulunmaktadır. Bu dosyalar, veri setine uygun olarak train ve test klasörlerine ayrılmıştır.
2. **Veri Temizleme:** Metin verileri genellikle gürültü içerir, bu da modelin doğru öğrenmesini engelleyebilir. Bu nedenle, verinin temizlenmesi gereklidir. Temizleme işlemi şunları içerir:
    - **Büyük/Küçük Harf Dönüşümü:** Tüm metinler küçük harfe dönüştürülür.
    - **Noktalamalar ve Sayılar:** Metinden gereksiz noktalamalar, sayılar ve semboller çıkarılır.
    - **Durdurma Kelimeleri (Stopwords):** Anlamsız olan ve sınıflandırma üzerinde etkisi olmayan kelimeler (örneğin "ve", "bir", "ile") metinden çıkarılır.
    - **Kelime Kökleri:** Anlamı bozmadan kelimelerin köklerine inmek için kelime kökleri bulunur ve bu kelimeler aynı köke indirgenir (lemmatization).

2.2. Farklı Vektörleştirme Tekniklerinin Açıklanması

Metin verilerini sayısal verilere dönüştürmek, makine öğrenmesi modellerinin metinleri işleyebilmesi için gereklidir. Bu dönüşümde kullanılan çeşitli vektörleştirme teknikleri bulunmaktadır. Bu teknikler, metni birer sayı dizisine (vektör) çevirerek modelin anlayabileceği hale getirir. Bu projede, dört farklı vektörleştirme yöntemi kullanılmıştır:

1. **Bag of Words (BoW):**
    - **Açıklama:** BoW, metindeki her kelimenin sıklığını dikkate alarak metni sayısal bir vektöre dönüştüren temel bir yöntemdir. Metnin her kelimesi bir özellik olarak kabul edilir ve her metin, bu kelimelerin sıklıklarıyla temsil edilir. BoW, metnin kelime sırasını dikkate almaz ve sadece kelimelerin varlığına ve sıklığına odaklanır.
    - **Uygulama:** Veri setindeki her kitap açıklaması, her kelimenin kaç kez geçtiğini gösteren bir vektörle temsil edilmiştir. Bu vektör, modelin metni anlayabilmesi için sayısal bir formatta oluşturulmuştur.
2. **TF-IDF (Term Frequency - Inverse Document Frequency):**
    - **Açıklama:** TF-IDF, kelimelerin metindeki sıklıklarının yanı sıra, kelimelerin genel önemini de dikkate alır. TF, bir kelimenin metindeki sıklığını, IDF ise kelimenin tüm metin koleksiyonundaki nadirliğini ölçer. Bu yöntem, metinlerde daha anlamlı ve nadir kelimelere daha fazla ağırlık verir.
    - **Uygulama:** TF-IDF yöntemi ile metinler, her kelimenin hem sıklığını hem de önemini gösteren vektörlere dönüştürülmüştür.

3. **Word2Vec:**
    - **Açıklama:** Word2Vec, kelimeleri sürekli değerli vektörlere dönüştüren bir modeldir. Bu model, kelimeler arasındaki anlamsal ilişkileri öğrenir ve benzer anlamlara sahip kelimeleri benzer vektörlerle temsil eder. Word2Vec, kelime bağlamını dikkate alarak anlamlı vektörler üretir.
    - **Uygulama:** Word2Vec modelini kullanarak her kelime için 100 boyutlu vektörler elde edilmiştir. Bu vektörler, kitap açıklamalarındaki kelimelerin anlamını yansıtır ve metinler, bu kelime vektörlerinin ortalamaları ile temsil edilmiştir.
4. **BERT (Bidirectional Encoder Representations from Transformers):**
    - **Açıklama:** BERT, bağlamı dikkate alarak kelimelerin anlamını öğrenen, son derece güçlü bir dil modelidir. BERT, her kelimenin hem sağındaki hem de solundaki kelimeleri dikkate alarak, her kelimenin daha zengin ve bağlamsal bir temsilini oluşturur.
    - **Uygulama:** BERT ile metinler, her bir kelime için derinlemesine anlamı yakalayacak şekilde 768 boyutlu vektörlerle temsil edilmiştir. Bu vektörler, kelimeler arasındaki ilişkiler ve metnin genel anlamı hakkında güçlü bir temsil sağlar.

3. Model Geliştirme

Bu aşamada, veri seti üzerinde farklı modeller eğitilerek performansları karşılaştırılmıştır. İki ana model türü kullanılmıştır: **MLP (Multilayer Perceptron)** ve **LSTM (Long Short-Term Memory)**. Her iki model de farklı vektörleştirme teknikleri ile beslenerek kitap türü tahminlerini yapmaktadır. Aşağıda, kullanılan model yapılarını ve hiperparametre optimizasyon tekniklerini açıklayacağız.

MLP Modeli

MLP (Multilayer Perceptron), sırasıyla bir giriş katmanı, birden fazla gizli katman ve çıkış katmanından oluşan bir yapay sinir ağı modelidir. Bu modelde, ilk katman 512 nöron ve ReLU aktivasyon fonksiyonu kullanarak verileri işler. İkinci katmanda 256 nöron ve yine ReLU aktivasyon fonksiyonu bulunur. Her iki gizli katmandan sonra dropout uygulanarak aşırı öğrenme (overfitting) engellenir. Çıkış katmanı, sınıf sayısına göre softmax aktivasyonu ile çok sınıflı sınıflandırma yapar. Modelin optimize edilmesinde **L2 düzenlilemesi** ve **Adam optimizer** kullanılır, böylece modelin genelleme yeteneği artırılır.

LSTM Modeli

LSTM (Long Short-Term Memory), sıralı verilerle çalışmak için tasarlanmış bir modeldir. Bu model, kelimelerin sıralarına duyarlı olup uzun dönemli bağımlılıkları öğrenebilir. Word2Vec veya BERT gibi vektörleştirme yöntemleri kullanıldığında, LSTM katmanları sırasıyla veriyi işler ve **Bidirectional LSTM** ile hem önceki hem de sonraki bağlamı dikkate alır. Modelde dropout uygulanarak aşırı öğrenme önlenir ve en son katmanda softmax aktivasyonu ile çok sınıflı sınıflandırma yapılır. **Adam optimizer** ve **categorical crossentropy** kayıp fonksiyonu ile modelin doğruluğu artırılır.

Model Performansı - MLP + BoW ve MLP + TF-IDF

MLP modeli ile BoW ve TF-IDF vektörleştirme yöntemleri kullanıldığında her iki model de mükemmel sonuçlar verdi. Hem BoW hem de TF-IDF yöntemleriyle yapılan tahminlerde doğruluk (accuracy) %100, precision, recall ve F1-score değerleri de 1.00 çıktı. Bu sonuçlar, modelin her sınıf için hatasız tahminler yaptığı anlamına geliyor.

Karşılaştırma ve Sınıflandırma Sonuçları

MLP + BoW ve MLP + TF-IDF modellerinin sınıflandırma raporlarına bakıldığında, her iki modelin de tüm sınıflarda (örneğin, 'horror', 'mystery', 'philosophy') 1.00 değerinde performans gösterdiği görüldü. Bu da modelin tüm etiketlerde doğru sınıflandırmalar yaptığını gösteriyor.

Vektörleştirme Yöntemlerinin Etkisi

BoW ve TF-IDF yöntemleri arasında herhangi bir performans farkı gözlemlenmedi. Her iki vektörleştirme tekniği de aynı yüksek başarıyı elde etti. Bu da BoW ve TF-IDF'nin veriyi çok iyi temsil ettiğini ve sınıflandırma için etkili olduğunu gösteriyor.

MLP + Word2Vec Sonuçları

MLP modelinin Word2Vec vektörleştirme yöntemiyle eğitildiği sonuçlar oldukça düşük performans gösterdi. Sadece 'religion' sınıfında %100 recall (hatırlama) elde edilse de, diğer sınıflarda model tamamen başarısız oldu. Precision, recall ve F1-score değerleri, çoğu sınıf için 0.00 çıktı. Sonuç olarak, Word2Vec bu modelde yetersiz kaldı ve genel doğruluk oranı %12 olarak belirlendi. Bu, modelin sınıfları doğru bir şekilde ayırt edemediğini gösteriyor.

MLP + BERT Sonuçları

MLP modelinin BERT vektörleştirme yöntemiyle eğitildiği sonuçlarda ise daha dengeli bir performans gözlemlendi. 'Horror', 'philosophy', 'sports' gibi bazı sınıflarda yüksek recall değerleri elde edilirken, genel doğruluk oranı %50 olarak belirlendi. Bununla birlikte, 'mystery', 'religion' ve 'romance' gibi bazı sınıflarda hala oldukça düşük değerler görüldü. F1-score ve precision değerleri de genel olarak düşük kaldı, fakat 'philosophy' ve 'sports' gibi sınıflarda belirli başarılar elde edildi. BERT'in sağladığı bilgileri bazı sınıflarda etkili bir şekilde kullandığı, ancak modelin hala tam verimli bir şekilde sınıflandırma yapamadığı söylenebilir.

Karşılaştırma

Word2Vec ve BERT vektörleştirme yöntemlerinin sonuçları birbirinden oldukça farklıydı. Word2Vec, modelin çoğu sınıfı doğru sınıflandıramaması nedeniyle çok düşük performans sergilerken, BERT bazı sınıflarda daha iyi sonuçlar gösterdi. Ancak her iki model de genel olarak BoW ve TF-IDF ile karşılaştırıldığında daha düşük doğruluk oranlarına sahipti. BERT, özellikle bazı sınıflarda daha iyi sonuçlar verirken, Word2Vec genel olarak başarısız oldu. Bu durum, BERT’in bağlam bilgilerini daha etkili şekilde kullanma potansiyeline sahip olduğunu ancak modelin hala geliştirilmesi gerektiğini gösteriyor.

MLP için Tüm Vektörleşme Yöntemlerinin Karşılaştırması ve Analizi

MLP + BoW ve MLP + TF-IDF, mükemmel sonuçlar vererek %100 doğruluk elde etti. Bu iki vektörleştirme tekniği, özellikle modelin küçük veri setinde çok başarılı olmasını sağladı. Ancak, Word2Vec ve BERT, daha düşük doğruluk oranları ile sınıflandırmada zorlandı. Word2Vec çok düşük bir doğruluk gösterirken, BERT belirli sınıflarda daha iyi sonuçlar elde etti ancak genel başarı oranı daha düşük kaldı. BERT’in bağlamsal analiz yetenekleri Word2Vec’e göre daha iyi olsa da, verilerin tam olarak anlaşılması ve modelin optimize edilmesi gerektiği açıkça görülüyor.

LSTM modellerinin karşılaştırmalı analizi

LSTM + BoW Sonuçları

LSTM + BoW modeli %100 doğruluk (accuracy) elde etti. Bu sonuç, BoW'nin basit yapısı ile LSTM modelinin sıralı verileri çok iyi öğrenebildiğini ve tüm sınıfları doğru bir şekilde tahmin ettiğini gösteriyor. BoW'nin bu tür veri setlerinde oldukça etkili olduğu görülüyor.

LSTM + TF-IDF Sonuçları

LSTM + TF-IDF modelinin doğruluk oranı %12.5 oldu. TF-IDF ile eğitilen model, verileri doğru sınıflandıramadı ve çoğu sınıf için sıfır performans sergiledi. Bu durum, TF-IDF'nin BoW kadar etkili olmadığı ve sıralı verilerde daha karmaşık bağlam bilgileri çıkaramadığı için LSTM'nin veriyi yeterince öğrenemediği anlamına geliyor.

LSTM + Word2Vec Sonuçları

LSTM + Word2Vec modelinde de benzer şekilde %12.5 doğruluk oranı elde edildi. Word2Vec, kelimelerin vektörel temsillerini sağlasa da, modelin çoğu sınıfı doğru tahmin edemediği ve performansın düşük olduğu gözlemlendi. Word2Vec'in bağlam ilişkisini anlamada bazı zorluklar yaşadığı ve LSTM ile yeterince güçlü sonuçlar veremediği söylenebilir.

LSTM + BERT Sonuçları

LSTM + BERT modelinin doğruluk oranı yine %12.5 oldu. BERT, dilin bağlamını iyi kavrayabilen bir model olmasına rağmen, burada yine düşük performans sergiledi. Bu sonuç, LSTM’nin BERT ile etkili bir şekilde entegre olamadığını ve modelin yeterince optimize edilmediğini gösteriyor.

Karşılaştırma ve Değerlendirme

LSTM + BoW, tüm sınıfları doğru tahmin ederek %100 doğruluk elde etti. Ancak, LSTM ile yapılan TF-IDF, Word2Vec ve BERT kombinasyonları çok düşük doğruluklar sergileyerek %12.5 gibi düşük bir performans gösterdi. Bu durum, LSTM modelinin basit vektörleştirme teknikleriyle (BoW) daha iyi performans gösterdiğini ve daha karmaşık bağlamları çıkaramayan vektörleştirme yöntemlerinin modelin başarısını düşürdüğünü ortaya koyuyor. Özellikle, TF-IDF, Word2Vec ve BERT'in sıralı veriyle etkili çalışamadığı veya modelin bu tekniklerle yeterince optimize edilmediği gözlemlendi.

Model ve Vektörleştirme Yöntemlerinin Karşılaştırılması: En İyi Sonuç

Karşılaştırmalı Sonuçlar ve Değerlendirme

Yukarıdaki doğruluk sonuçlarına bakıldığında, en iyi performansı MLP + BoW modeli elde etti. Ancak, hem eğitim hem de test setindeki doğruluk değerleri oldukça düşük. Test doğruluğu %12.5 ile sınırlı olsa da, bu model diğerlerine göre bir adım önde görünüyor diyebiliriz.

Diğer tüm modeller, MLP + TF-IDF, MLP + Word2Vec, MLP + BERT, ve LSTM + tüm vektörleştirme yöntemleri (BoW, TF-IDF, Word2Vec, BERT) %12.5 doğrulukla sonuçlandı. Bu da, modellerin genel olarak yeterli öğrenmeyi sağlayamadığını ve doğru tahminlerde bulunmakta zorlandığını gösteriyor.

En İyi Model: MLP + BoW

Sonuçları değerlendirdiğimizde, MLP + BoW modelinin, diğer tüm kombinasyonlardan daha iyi bir doğruluk elde ettiği söylenebilir. Ancak, hem eğitim hem de test doğruluklarının düşük olması, modelin iyileştirilmesi gerektiğini gösteriyor. Diğer tüm model kombinasyonları ise benzer şekilde düşük performans sergileyerek %12.5 doğruluk ile sonuçlandı.

Sonuç ve Öneriler:

BoW'nin, veri setine uygun ve basit bir yaklaşım olduğu görülmektedir. Diğer vektörleştirme yöntemleri (TF-IDF, Word2Vec, BERT), verinin bağlamını yeterince çıkaramamış olabilir ya da bu modellerin hiperparametreleri, öğrenme sürecinde yeterince optimize edilemedi diyebiliriz. Sonuç olarak, MLP + BoW modelinin şu an için en iyi performansı sergilediğini söyleyebiliriz, ancak model iyileştirmeleri ve farklı hiperparametre ayarları ile başarı artırılabilir.

MLP Modelinde Yapılan İyileştirmeler ve Denemeler

1. Dropout Artırılması ve L2 Regularization Ekleme

- Dropout oranı 0.3'ten 0.5'e çıkarıldı ve L2 regularization eklendi.
- **Sonuçlar**:
    - Genel doğruluk arttı: %75 → %88
    - Macro ve weighted F1-score yükseldi: 0.69 → 0.83
    - Horror ve Romance sınıflarının doğruluğu %0'dan %1'e çıktı.
    - Ancak, Science-fiction sınıfı başarısız oldu.
- **Yorum**: Bu iyileştirme, overfitting'i azalttı ve modelin daha dengeli tahminler yapmasını sağladı. Science-fiction sınıfını iyileştirmek için veri artırma, class weight veya batch size gibi yöntemlerle iyileştirme yapılabilir.

2. Batch Size Denemeleri

- Batch size, 32'den 64'e çıkarıldı ve sonuçlar değerlendirildi.
    - Batch size 64 modelin öğrenme yeteneğini düşürdü. Küçük veri setlerinde küçük batch size (16-32) daha iyi çalışır.
- **Sonuç**: Batch size 16, bazı sınıflar için iyi çalışsa da modelin başarısını düşürdü. Batch size 32 daha dengeli bir sonuç sağladı.

3. Learning Rate Ayarları

- Learning rate 0.001'den 0.0005'e düşürüldü: Modelin genel performansı kötüleşti. Test doğruluğu %50'ye düştü.
- Learning rate 0.0008'e çıkarıldı: Bazı sınıflarda iyileşmeler oldu, ancak genel doğruluk %88'den %75'e düştü.
- **Yorum**: Çok düşük learning rate (0.0005) modelin yeterince öğrenememesine yol açtı. 0.0008 ise, 0.001 kadar etkili değil ancak daha iyi sonuçlar sundu.

4. Epoch Sayısı Artırma ve Overfitting

- Epoch sayısı 10'dan 30'a çıkarıldı.
    - **Sonuç**: 30 epoch'ta overfitting gözlemlendi, Train doğruluğu %100'e ulaşırken, Test doğruluğu %62.5'e düştü.
    - **Yorum**: Epoch sayısını arttırmak modelin öğrenmesini sağladı ancak aşırı öğrenme ve overfitting sorununu ortaya çıkardı.

5. Dropout ve L2 Regularization ile Modeli İyileştirme

- Dropout oranı 0.6 ve L2 regularization ile model yeniden eğitildi.
    - **Sonuç**: Test doğruluğu %100'e çıktı ancak overfitting gözlemlendi. Model, eğitim verisini mükemmel öğrenmiş ancak test verisine genelleme yapamıyordu.

6. Model Boyutunu Küçültme (Nöron Sayısı)

- Model boyutu küçültüldü: 512 → 256 → 128 nöron.
    - **Sonuç**: Bu, overfitting'i azalttı ancak model yeterince öğrenemedi. Test doğruluğu %62.5'e düştü.
    - **Yorum**: Model çok küçük olduğu için underfitting sorunu yaşandı, bu da öğrenme kapasitesini azalttı.
    -

MLP + BERT Denemeleri

1. Epoch Sayısını Arttırma

- Epoch sayısı 10'dan 20'ye çıkarıldı.
    - **Sonuç**: Test doğruluğu %12'den %62'ye çıktı.
    - **Yorum**: Epoch sayısının artırılması, modelin iyileşmesini sağladı ancak dengesizlikler devam etti.

2. Early Stopping ile Overfitting'i Azaltma

- Early stopping eklendi ve epoch 6'da durduruldu.
    - **Sonuç**: Test doğruluğu %62.5'e çıktı.
    - **Yorum**: Early stopping, modelin gereksiz eğitiminin önüne geçti ve overfitting'i azalttı.

3. Batch Size Değişiklikleri

- Batch size 16'dan 64'e çıkarıldı.
    - **Sonuç**: Test doğruluğu %62.5'ten %75'e çıktı.
    - **Yorum**: Batch size'ı büyütmek modelin daha stabil öğrenmesini sağladı, ancak train doğruluğu %100'de kaldı.

LSTM Modeli İyileştirmeleri

1. BoW ve LSTM Uyumsuzluğu

- **Sonuçlar**: BoW, sırasız bir vektörleştirme yöntemi olduğu için LSTM ile uyumsuzdu. Performans iyileşmesi sağlanamadı.

2. Batch Size Denemeleri

- Batch size 16, 32 ve 64 olarak test edildi.

- **Sonuç**: Batch size 32 en dengeli sonuçları verdi, diğerleri ise modelin doğruluğunu düşürdü.

3. Epoch Sayısı Artırma

- Epoch sayısı 10'dan 30'a çıkarıldı.
    - **Sonuç**: Epoch sayısını artırmak, öğrenmeyi artırdı ancak overfitting meydana geldi. Daha fazla epoch kullanmak için overfitting'i önleyici önlemler alınmalı diyebilirizç.

Genel Değerlendirme ve Öneriler:

- **En iyi model**: MLP + BoW kombinasyonu genel olarak en iyi doğruluk oranlarına ulaştı.
- **En iyi vektörleştirme yöntemi**: BoW, LSTM ile uyumsuz olsa da, MLP ile en iyi sonuçları verdi.
- **Overfitting'i azaltmak için neler yapabiliriz**: Batch size, epoch sayısı ve dropout oranları üzerinde dikkatli ayarlamalar yapılabilir. Early stopping ve learning rate ayarları da performans iyileştirmesinde önemli rol oynamaktadır.

En İyi Performans Gösteren Yaklaşım:

- **Model**: MLP (Multilayer Perceptron)
- **Vektörleştirme Yöntemi**: BoW (Bag of Words)

Nedenler:

- MLP + BoW kombinasyonu, en yüksek doğruluk oranlarına (Train: %88, Test: %75) ulaştı.
- **BoW**, basit ve etkili bir yöntem olarak, sırasız veri içeren metinlerin işlenmesinde iyi çalıştı ve MLP modelinin, verilen metin açıklamalarına göre doğru sınıflandırmalar yapmasına yardımcı oldu.
- Diğer vektörleştirme yöntemleri (TF-IDF, Word2Vec, BERT) bazı sınıflarda iyileşmeler sağlasa da genel olarak **BoW + MLP** kombinasyonu daha stabil sonuçlar verdi.

Gelecekteki İyileştirmeler İçin Öneriler:

- **Veri Artırma**: Özellikle bazı sınıflar için veri sayısını artırarak modelin bu sınıflara olan duyarlılığını artırabiliriz.
- **Model Optimizasyonu**: Learning rate ve batch size gibi hiperparametrelerin daha hassas bir şekilde ayarlanması, daha stabil ve doğru sonuçlar elde edebiliriz.
- Birden fazla modelin çıktılarının birleştirildiği yöntemleri (örneğin, Random Forest veya XGBoost) denenebilir. Bu yöntem, modelin genelleme yapma yeteneğini artırabilir.