



YAPAY ZEKA PROJE ÖDEVİ MÜZİK TÜRÜ SINIFLANDIRMA

Mert Tuna Kurnaz
20011031
tuna.kurnaz@std.yildiz.edu.tr

Büşra Medine Gural
20011038
medine.gural@std.yildiz.edu.tr

1. GİRİŞ

Müzik türü sınıflandırma, bir müzik parçasının otomatik olarak belirli bir türe atanmasını amaçlayan bir makine öğrenimi problemidir. Bu proje, çeşitli müzik dosyalarından elde edilen ses özelliklerini kullanarak müzik türlerini sınıflandırmayı amaçlamaktadır. Bu bağlamda, farklı makine öğrenimi ve derin öğrenme modelleri kullanılarak bir dizi deney gerçekleştirilmiştir.

2. GELİŞTİRME SÜRECİNDE YAŞANANLAR

2.1 Veri Seti

Proje kapsamında kullanılan "GTZAN Music Genre Dataset [1]", her biri 30 saniye uzunluğunda olan 10 farklı müzik türünden 100'er adet ses dosyasını içermektedir. Bu veri seti, müzik türü sınıflandırma projeleri için yaygın olarak kullanılan bir referans veri setidir. Jazz klasöründeki 54. müzik dosyası bozuk olduğu için bu dosya silinmiştir. Türler arasında blues, classical, country, disco, hiphop, jazz, metal, pop, reggae ve rock bulunmaktadır. Bu çeşitlilik, modelin farklı müzik türlerini ayırt etme yeteneğini test etmek için uygundur.

2.2 Özellik Çıkarımı

Ses dosyalarından çeşitli özellikler çıkarılarak modellerin öğrenmesi için uygun bir veri kümesi oluşturulmuştur. Bu özellikler şunlardır:

- **Mel-frekans kepstrum katsayıları (MFCCs):** İnsan kulağının algıladığı şekilde sesin frekans içeriğini temsil eder. Sesin temel karakteristik özelliklerini yakalar.
- **Kromagram (Chroma):** Müzikal akorların ve tonların temsilini sağlar. Sesin harmonik yapısını belirler.
- **Mel spektrogram:** Sesin zaman içindeki frekans içeriğini gösterir. Frekans bileşenlerinin zamanla nasıl değiştiğini gözlemlemeye olanak tanır.
- **Spektral kontrast:** Farklı frekans bantları arasındaki kontrastı ölçer. Müziğin spektral özelliklerini analiz eder.
- **Spektral roll-off:** Enerjinin %85'inin bulunduğu frekans noktasıdır. Sesin enerji dağılımını temsil eder.
- **Sıfır geçiş oranı (Zero-crossing rate):** Sinyalin sıfır noktasını geçme sayısını ölçer. Gürültü ve periyodik olmayan bileşenlerin varlığını belirler.
- **Kök ortalama kare enerji (Root Mean Square Energy - RMSE):** Sinyalin enerji içeriğini ölçer. Sesin genel güç seviyesini temsil eder.

Bu özellikler, ses dosyalarının frekans ve zaman alanındaki temsilini sağlayarak, müzik türü sınıflandırma modellerinin daha etkin bir şekilde öğrenmesine olanak tanımaktadır. Özellik çıkarımı için **librosa** kütüphanesi kullanılmıştır.

2.3 Model Eğitimi ve Değerlendirme

Proje kapsamında, iki farklı makine öğrenimi algoritması ve bir derin öğrenme modeli kullanılmıştır:

- K-Nearest Neighbors (KNN): KNN, en yakın komşulara dayalı olarak sınıflandırma yapar. Bu model, basit ve yorumlanabilir olması nedeniyle tercih edilmiştir. Modelin performansı, hiperparametre optimizasyonu ile artırılmıştır. En iyi parametreler Grid Search yöntemi kullanılarak belirlenmiştir.
 - Hiperparametreler: Komşu sayısı (k), ağırlıklandırma (uniform, distance), mesafe metriği (euclidean, manhattan, minkowski)
 - Performans Değerlendirme: 5-fold çapraz doğrulama kullanılarak doğruluk, kesinlik, duyarlılık ve F1 skoru gibi metrikler hesaplanmıştır.
- Random Forest (RF): RF, birden fazla karar ağacından oluşan ve bu ağaçların çoğunluk oyuna göre sınıflandırma yapan bir topluluk yöntemidir. Bu model, veri setindeki çeşitliliği ve karmaşıklığı daha iyi yakalayabilir.
 - Hiperparametreler: Ağaç sayısı (n_estimators), maksimum derinlik (max_depth), minimum örnek split sayısı (min_samples_split), minimum yaprak sayısı (min_samples_leaf), maksimum özellik sayısı (max_features)
 - Performans Değerlendirme: 5-fold çapraz doğrulama ve Grid Search yöntemi kullanılarak en iyi hiperparametreler belirlenmek istenmiştir ancak alınan hatadan dolayı parametreler statik olarak verilmiştir. Modelin doğruluk metriği test seti üzerinde değerlendirilmiştir.
- Yapay Sinir Ağları (YSA): Derin öğrenme modeli olarak bir yapay sinir ağı kullanılmıştır. Bu model, daha karmaşık ve yüksek boyutlu verilerle çalışabilme yeteneğine sahiptir.
 - Model Mimarisi: İki katmanlı tam bağlantılı (Dense) katmanlar, Dropout katmanları ile kullanılmıştır. Aktivasyon fonksiyonları olarak ReLU ve softmax kullanılmıştır.
 - Eğitim: Model, 100 epoch boyunca eğitim verilmiştir. Adam optimizyer kullanılarak eğitim süreci hızlandırılmış ve doğruluk metriği izlenmiştir.
 - Performans Değerlendirme: Modelin doğruluğu test seti üzerinde ölçülmüştür.

Her bir model, veri setinden çıkarılan özellikler kullanılarak eğitilmiş ve performansları değerlendirilmiştir. Ayrıca, modellerin hiperparametre ayarları Grid Search yöntemi ile optimize edilmiştir.

3. ÇALIŞTIRMA ÖRNEKLERİ

“eminem.wav” için KNN ve Random Forest sonucu:

```
[89] import librosa

#external_audio_path = "drive/MyDrive/Data/genres_original/blues/blues.00001.wav"
external_audio_path = "drive/MyDrive/Data/eminem.wav"

features = extract_features(external_audio_path) #extract features from test audio

# input for the model
input_features = np.array(features['mfccs_mean'] + features['chroma_mean'] + features['mel_spectrogram_mean'] +
                           features['spectral_contrast_mean'] + features['spectral_rolloff_mean'] +
                           features['zero_crossing_rate_mean'] + features['rmse_mean']).reshape(1, -1)

new_sample_scaled = scaler.transform(input_features)
predicted_class = knn_model.predict(new_sample_scaled)
print("KNN Predicted:", predicted_class)

predict = rf_custom.predict(new_sample_scaled)
print("RF Predicted:", predict)
```

KNN Predicted: ['pop']
RF Predicted: ['pop']

“eminem.wav” için Yapay Sinir Ağları sonucu:

```
external_audio_path = "drive/MyDrive/Data/eminem.wav"

features = extract_features(external_audio_path)

input_features = np.array(features['mfccs_mean'] + features['chroma_mean'] + features['mel_spectrogram_mean'] +
                           features['spectral_contrast_mean'] + features['spectral_rolloff_mean'] +
                           features['zero_crossing_rate_mean'] + features['rmse_mean']).reshape(1, -1)

input_features_scaled = scaler.transform(input_features)

input_features_scaled = input_features_scaled.reshape((-1, input_features_scaled.shape[1]))

predicted_class = model.predict(input_features_scaled)
predicted_genre = encoder.inverse_transform(np.argmax(predicted_class, axis=1))

print(f"Predicted Genre: {predicted_genre[0]}")
```

1/1 [=====] - 0s 33ms/step
Predicted Genre: pop

“class.wav” için KNN ve Random Forest sonucu:

```
import librosa

#external_audio_path = "drive/MyDrive/Data/genres_original/blues/blues.00001.wav"
external_audio_path = "drive/MyDrive/Data/class.wav"

features = extract_features(external_audio_path) #extract features from test audio

# input for the model
input_features = np.array(features['mfccs_mean'] + features['chroma_mean'] + features['mel_spectrogram_mean'] +
                           features['spectral_contrast_mean'] + features['spectral_rolloff_mean'] +
                           features['zero_crossing_rate_mean'] + features['rmse_mean']).reshape(1, -1)

new_sample_scaled = scaler.transform(input_features)
predicted_class = knn_model.predict(new_sample_scaled)
print("KNN Predicted:", predicted_class)

predict = rf_custom.predict(new_sample_scaled)
print("RF Predicted:", predict)
```

KNN Predicted: ['classical']
RF Predicted: ['classical']

“class.wav” için Yapay Sinir Ağları sonucu:

```
external_audio_path = "drive/MyDrive/Data/class.wav"

features = extract_features(external_audio_path)

input_features = np.array(features['mfccs_mean'] + features['chroma_mean'] + features['mel_spectrogram_mean'] +
                           features['spectral_contrast_mean'] + features['spectral_rolloff_mean'] +
                           features['zero_crossing_rate_mean'] + features['rmse_mean']).reshape(1, -1)

input_features_scaled = scaler.transform(input_features)

input_features_scaled = input_features_scaled.reshape((-1, input_features_scaled.shape[1]))

predicted_class = model.predict(input_features_scaled)
predicted_genre = encoder.inverse_transform(np.argmax(predicted_class, axis=1))

print(f"Predicted Genre: {predicted_genre[0]}")
```

1/1 [=====] - 0s 33ms/step
Predicted Genre: classical

4. YORUMLAR

Projenin başlangıç aşamasında, veri setinden özellik çıkarma işlemi librosa kütüphanesi kullanılarak gerçekleştirilmiştir. Bu süreç, 999 müzik dosyası için 7 farklı özelliğin çıkarılmasını içermekte olup, oldukça zaman alıcı olmuştur. Özellik çıkarma işlemi, her bir ses dosyasının frekans ve zaman alanındaki temsilini sağlayarak, sınıflandırma modellerinin etkinliğini artırmayı hedeflemiştir.

Proje boyunca elde edilen sonuçlar değerlendirildiğinde, kullanılan farklı modellerin performanslarının çeşitli düzeylerde olduğu gözlemlenmiştir.

- K-Nearest Neighbors (KNN) modeli, %65.5 doğruluk oranı ile müzik türü sınıflandırma görevinde makul bir performans sergilemiştir. KNN modelinin düşük doğruluk oranı, modelin basit yapısından ve veri setinin şeklinden kaynaklanıyor olabilir.
- Random Forest (RF) modeli, %69.5 doğruluk oranı ile KNN modelinden daha iyi bir performans göstermiştir. Random Forest modeli, daha fazla esneklik ve çeşitlilik sağladığı için daha yüksek doğruluk oranlarına ulaşmıştır. Bununla birlikte, RF modelinin hiperparametrelerinin Grid Search ile en iyi hale ulaşması sonucunda daha iyi sonuç elde edilebileceği düşünülmektedir.
- Yapay Sinir Ağları (YSA) modeli, %69 doğruluk oranı ile performans gösteren model olmuştur. Özellikle veri setinin boyutuna ve çeşitliliğine rağmen, derin öğrenme modelinin karmaşıklığı ve öğrenme kapasitesi sayesinde daha yüksek doğruluk oranına ulaşmıştır. Ancak, YSA modeli, eğitim süresi ve hesaplama maliyeti açısından daha zorlayıcı olmuştur.

Bu sonuçlar, Random Forest ve Yapay Sinir Ağları modellerinin müzik türü sınıflandırma görevinde daha etkili olduğunu göstermektedir. Gelecekte, daha büyük ve çeşitli veri setleri kullanılarak bu modellerin performansını daha da artırmak mümkündür. Ayrıca, daha ileri özellik çıkarma teknikleri ve hiperparametre optimizasyonları ile model performansları iyileştirilebilir.

Dipnot: Videoda yapay sinir ağlarından yanlışlıkla CNN olarak bahsedilmektedir. Kodun önceki versiyonundan dolayı bu hata yapılmıştır.

5. SAYISAL BAŞARI

- K-Nearest Neighbors (KNN)
 - Doğruluk (Accuracy): %44.7
 - Kesinlik (Precision - Weighted): %47.6
 - Duyarlılık (Recall - Weighted): %44.7
 - F1 Skoru (F1 - Weighted): %44.9
 - Test Seti Doğruluğu: %65.5
- Random Forest (RF)
 - Test Seti Doğruluğu: %69.5
- Yapay Sinir Ağları (YSA)
 - Test Seti Doğruluğu: %69.0

Başarıyı ölçmek için doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru gibi metrikler kullanılmıştır. K-fold çapraz doğrulama (cross-validation) yöntemi kullanılarak modellerin performansı değerlendirilmiştir. Her bir modelin farklı müzik türlerini ne kadar doğru sınıflandırabildiği analiz edilerek, modellerin performansları karşılaştırılmıştır.

6. KAYNAKLAR

- [1] <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification/data>
- <https://dergipark.org.tr/en/download/article-file/2790248>
- https://www.researchgate.net/publication/356377974_Music_Genre_Classification_Techniques
- <https://medium.com/@namratadutt2/music-genre-classification-using-cnn-part-1-feature-extraction-b417547b8981>
- <https://dergipark.org.tr/en/download/article-file/3482271>