 **T.C.**

**SİVAS CUMHURİYET ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**SES İŞLEME DERSİ VİZE ÖDEVİ**

**MFCC - LOGISTIC REGRESSION**

**YAPAY ZEKA VE VERİ BİLİMİ**

**ANA BİLİM DALI**

**SİVAS-2024**

**NİLBERK AYBÜKE ÖZDEMİR - 20239258001**

**MFCC LOGISTIC REGRESSION İŞLEMLERİ**

**MFCC Özellikleri Çıkartma ve Logistic Regression Modeli Eğitme**

Bu yazıda, ses dosyalarından MFCC (Mel-Frekans Kepstral Katsayıları) özelliklerini çıkartarak bir Logistic Regression modeli eğitip test edeceğiz. Ayrıca, modelin performansını iyileştirmek için Grid Search ile hiper-parametre optimizasyonu yapacağız.

**Kullanılan Kütüphaneler**

Öncelikle, gerekli kütüphaneleri içe aktarıyoruz:

import numpy as np

from glob import glob

from scipy.io import wavfile

from python\_speech\_features import mfcc

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import LabelEncoder, StandardScaler

**MFCC Özelliklerini Çıkartma**

Ses dosyalarından MFCC özelliklerini çıkartmak için bir fonksiyon yazıyoruz. MFCC, sesin spektral özelliklerini temsil eden özniteliklerdir ve genellikle ses tanıma uygulamalarında kullanılır.

def extract\_features(file\_path):

rate, audio = wavfile.read(file\_path) # Ses dosyasını okuyoruz

mfcc\_features = mfcc(audio, rate, numcep=13) # MFCC özelliklerini çıkartıyoruz

return mfcc\_features.flatten() # Özellikleri tek bir diziye düzlüyoruz

**Veri Setini Yükleme**

Veri setindeki tüm ses dosyalarını okuyup MFCC özelliklerini çıkartan ve etiketleri belirleyen bir fonksiyon yazıyoruz.

def load\_data(data\_path):

features = []

labels = []

for class\_dir in glob(data\_path + "/\*"): # Her bir sınıf için

for audio\_file in glob(class\_dir + "/\*.wav"): # Her bir ses dosyası için

features.append(extract\_features(audio\_file)) # MFCC özelliklerini çıkartıyoruz

labels.append(class\_dir.split("/")[-1]) # Sınıf etiketini belirliyoruz

return np.array(features), np.array(labels)

**Veri Setini Yükleme ve Hazırlama**

Veri setinin yolunu belirleyip yükleme fonksiyonunu çağırıyoruz.

data\_path = "../data/TurEV-DB-master/TurEV-DB-master/Sound Source/"

X, y = load\_data(data\_path)

Etiketleri sayısal değerlere dönüştürüyoruz.

label\_encoder = LabelEncoder()

y = label\_encoder.fit\_transform(y)

Veriyi eğitim ve test setlerine ayırıyoruz.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

Veriyi standart ölçeklendiriyoruz.

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Modeli Oluşturma ve Eğitme**

Logistic Regression modelini oluşturup eğitim verisi ile eğitiyoruz.

model = LogisticRegression(max\_iter=1000)

**Grid Search ile Hiper-Parametre Optimizasyonu**

Modelin performansını iyileştirmek için Grid Search kullanarak en iyi hiper-parametreleri belirliyoruz.

param\_grid = {

'C': [0.01, 0.1, 1, 10, 100], # Düzenleme parametresi

'solver': ['newton-cg', 'lbfgs', 'liblinear'] # Çözümleyiciler

}

grid\_search = GridSearchCV(model, param\_grid, cv=5, n\_jobs=-1, scoring='accuracy')

grid\_search.fit(X\_train, y\_train)

En iyi parametreleri ve skoru yazdırıyoruz.

print("En iyi parametreler:", grid\_search.best\_params\_)

print("En iyi skor (doğruluk):", grid\_search.best\_score\_)

**Modelin Performansını Değerlendirme**

En iyi model ile eğitim ve test doğruluklarını değerlendiriyoruz.

best\_model = grid\_search.best\_estimator\_

train\_accuracy = best\_model.score(X\_train, y\_train)

test\_accuracy = best\_model.score(X\_test, y\_test)

print("Eğitim doğruluğu:", train\_accuracy)

print("Test doğruluğu:", test\_accuracy)

**Sonuç**

Bu kod, ses dosyalarından MFCC özelliklerini çıkartıp bir Logistic Regression modeli eğitiyor ve test ediyor. Modelin performansını iyileştirmek için Grid Search ile hiper-parametre optimizasyonu yapılıyor. En iyi parametrelerle modelin eğitim ve test doğrulukları değerlendirilerek performansı ölçülüyor. Bu sayede, ses tanıma görevlerinde kullanılabilecek etkili bir sınıflandırma modeli oluşturulmuş oluyor.