

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



AKUSTİK ÖZELLİKLER KULLANARAK MÜZİK ÖNERİ VE
SINIFLANDIRMA SİSTEMİ

17011080 – Engin Deniz ÇAĞLAR
18011607 – Ahmet YAMAN

BİLGİSAYAR PROJESİ

Danışman
Dr. Ahmet ELBİR

Haziran, 2021

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde değerli bilgilerini bizimle paylaşan, kendisine danıştığımız her alanda kıymetli zamanını içtenlikle ve büyük sabırla bize veren, her türlü problemde yardım alabildiğimiz değerli danışmanımız Dr. Ahmet Elbir'e teşekkürü bir borç biliyor ve şükranlarımızı sunuyoruz.

Engin Deniz ÇAĞLAR
Ahmet YAMAN

İÇİNDEKİLER

SİMGE LİSTESİ	vi
KISALTMA LİSTESİ	vii
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ	ix
ÖZET	x
ABSTRACT	xi
1 Giriş	1
2 Ön İnceleme	3
3 Fizibilite	5
3.1 Teknik Fizibilite	5
3.1.1 Yazılım Fizibilitesi	5
3.1.2 Donanım Fizibilitesi	6
3.2 Zaman Fizibilitesi	6
3.3 Yasal Fizibilite	6
4 GTZAN Veri Seti	7
4.1 GTZAN Veri Seti Nedir?	7
4.2 Müzik Türü Sınıflandırma Yaklaşımları:	8
4.3 Özellik çıkarma	9
4.4 Özet:	9
5 Özellik Çıkarımı	10
5.1 Spektral Ağırlık Merkezi (Spectral Centroid)	10
5.2 Spektral Bant Genişliği (Spectral Bandwidth)	10
5.3 Spektral Kontrast (Spectral Contrast)	11
5.4 Spektral Düzlük (Spectral Flatness)	11
5.5 Spektral Dönüm Noktası (Spectral Rolloff)	12

5.6	Sıfır Geçiş Oranı (Zero Crossing Rate)	12
5.7	Polinom Özellikleri (Polynomial Features)	13
5.8	Karakök Ortalama Karelerin Enerjisi (Root Mean Square Energy)	14
5.9	Mel-Frekansı Cepstral Katsayıları (Mel-Frequency Cepstral Coefficients)	14
5.10	Kroma Sabit-Q Dönüşümü (Chroma Constant-Q Transform)	15
6	Boyut İndirgeme ve Özellik Seçim Yöntemleri	16
6.1	Temel Bileşen Analizi (Principal Component Analysis)	16
6.2	İleri Seçim (Forward Selection)	16
6.2.1	P-Value Değeri	16
7	Makine Öğrenmesi Algoritmaları	18
7.1	Lojistik Regresyon (Logistic Regression)	18
7.2	K En Yakın Komşu (K-Nearest Neighbors)	18
7.3	Destek Vektör Makinesi (Support Vector Machine)	19
7.3.1	Çekirdek Hilesi (Kernel Trick)	20
7.4	Naive Bayes Sınıflandırıcısı (Naive Bayes)	20
7.5	Karar Ağacı (Decision Tree)	21
7.6	Rassal Orman (Random Forest)	22
7.7	Quadratic Diskriminant (Quadratic Discriminant)	22
7.8	Doğrusal Diskriminant (Linear Discriminant)	23
7.9	Gauss Süreç Regresyonu (Gaussian Process Regression)	23
7.10	Sinir Ağı (Neural Network)	24
7.11	Uyarlanabilir Güçlendirme (AdaBoost)	24
8	Model Performansı Değerlendirme Yöntemleri	25
8.1	Karmaşıklık Matrisi (Confusion Matrix)	25
8.2	Doğruluk (Accuracy)	26
8.3	Keskinlik (Precision)	26
8.4	Duyarlılık (Recall)	26
8.5	F1 Skor	27
8.6	Cohen's Kappa	27
9	Deneysel Sonuçlar	28
9.1	Veri Kümesiyle İşlemler	28
9.2	Öznitelik Çıkarımı Sonrası Sonuçlar	29
9.2.1	En İyi Sonuçlar	30
9.3	Özniteliklerin Doğruluk Skoruna Etkileri	31
9.4	Temel Bileşen Analizi ile Sonuçlar	33
9.4.1	Dağılım Grafiği	33

9.4.2	Algoritma Sonuçları	33
9.4.3	En İyi Sonuçlar	35
9.5	İleri Seçim ile Sonuçlar	37
9.5.1	Algoritma Sonuçları	37
9.5.2	En İyi Sonuçlar	38
10	Sonuç	40
	Referanslar	42
	Özgeçmiş	45

SİMGE LİSTESİ

κ	Cohen's Kappa
f	Frekans
C	Kovaryans Matrisi
β	Lineer Model Katsayıları
M	Mel Ölçeği
μ	Ortalama Vektörler

KISALTMA LİSTESİ

CQT	Constant-Q Transform
FN	False Negative
FP	False Positive
GPR	Gaussian Process Regression
GB	GigaByte
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
MFCC	Mel-Frequency Cepstral Coefficients
PCA	Principal Component Analysis
RMSE	Root Mean Square Energy
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
ZCR	Zero Crossing Rate

ŞEKİL LİSTESİ

Şekil 3.1	Gantt Çizelgesi	6
Şekil 4.1	Sınıflandırma Süreci	7
Şekil 5.1	Spektral Ağırlık Merkezi	10
Şekil 5.2	Spektral Bant Genişliği	11
Şekil 5.3	Spektral Kontrast	11
Şekil 5.4	Spektral Düzlük	12
Şekil 5.5	Spektral Dönüm Noktası	12
Şekil 5.6	Sıfır Geçiş Oranı	13
Şekil 5.7	Polinom Özellikleri	13
Şekil 5.8	Mel-Frekans Cepstrumu	14
Şekil 5.9	Sabit-Q Dönüşümü	15
Şekil 7.1	Lojistik Regresyon	18
Şekil 7.2	K En Yakın Komşu	19
Şekil 7.3	Destek Vektör Makinesi	19
Şekil 7.4	Çekirdek Hilesi	20
Şekil 7.5	Çekirdek Hilesi 2. Boyuttan 3. Boyuta	20
Şekil 7.6	Karar Ağacı	21
Şekil 7.7	Rassal Orman	22
Şekil 7.8	İkinci Dereceden Sınıflandırıcı	22
Şekil 7.9	Doğrusal Diskriminant	23
Şekil 7.10	Gauss Süreç Regresyonu	23
Şekil 7.11	Sinir Ağı	24
Şekil 7.12	Uyarlanabilir Güçlendirme	24
Şekil 8.1	Karmaşıklık Matrisi	25
Şekil 9.1	Karmaşıklık Matrisi	31
Şekil 9.2	Öznitelik - Doğruluk Grafiği	32
Şekil 9.3	Öznitelik - Doğruluk Grafiği	32
Şekil 9.4	Üçüncü Veri Kümesi PCA	33
Şekil 9.5	Karmaşıklık Matrisi	36
Şekil 9.6	Karmaşıklık Matrisi	39

TABLO LİSTESİ

Tablo 3.1	Donanım Bilgileri	6
Tablo 8.1	Cohen's Kappa	27
Tablo 9.1	En İyi Skorlar	30
Tablo 9.2	En İyi Parametreler	30
Tablo 9.3	En İyi Sonuçlar	31
Tablo 9.4	PCA ile En İyi Skorlar	34
Tablo 9.5	PCA ile En İyi Parametreler	35
Tablo 9.6	PCA ile En İyi Sonuçlar	35
Tablo 9.7	İleri Yönde Seçim En İyi Skorlar	38
Tablo 9.8	İleri Yönde Seçim En İyi Parametreler	38
Tablo 9.9	İleri Yönde Seçim En İyi Sonuçlar	39

Akustik Özellikler Kullanarak Müzik Öneri ve Sınıflandırma Sistemi

Engin Deniz ÇAĞLAR

Ahmet YAMAN

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Dr. Ahmet ELBİR

Modern yaşamın gelişmesi ile birlikte teknolojiye olan ihtiyaç her geçen gün artmaktadır. Her alanda olduğu gibi bu teknolojik yeniliklere ayak uydurabilmek de insanlık için büyük önem arz etmektedir. Teknoloji, beraberinde verileri meydana getirmektedir. Bu bilgileri işleyip anlamlı hale getirmek de makine öğrenmesi algoritmalarıyla sağlanabilmektedir.

Bu çalışmada makine öğrenmesi yöntemleriyle müzik türü sınıflandırma sistemi üzerinde çalışıldı. Bu kapsamda sayısal işaret işleme teknikleri kullanarak GTZAN veri setindeki ses dosyalarından zaman frekans analizi yöntemlerini kullanıp özellikler çıkartıldı. Bu sürecin ardından elde edilen özellikler sınıflandırma yöntemlerinde kullanıldı. Boyut indirgeme ve öznitelik seçimi yöntemleri kullanılarak sonuçlar karşılaştırıldı.

Sonuç olarak geldiğimiz noktada Sinir Ağı algoritması sayesinde %72 başarı oranına ulaşıldı.

Anahtar Kelimeler: Makine öğrenmesi algoritmaları, müzik öneri sistemi, sınıflandırma sistemi, akustik özellik kullanımı, ses, müzik, sinyal işleme, spektral özellikler, MFCC, CHROMA CQT

Music Recommendation and Classification System Using Acoustic Properties

Engin Deniz ÇAĞLAR

Ahmet YAMAN

Department of Computer Engineering
Computer Project

Advisor: Asst. Prof. Ahmet ELBİR

With the development of modern life, the need for technology is increasing day by day. As in every field, keeping up with these technological innovations is great importance for humanity. Technology creates data with it. Processing this information and making it meaningful can also be achieved with machine learning algorithms.

In this study, the music genre classification system was studied with machine learning methods. In this context, by using digital signal processing techniques, features were extracted from the audio files in the GTZAN dataset using time-frequency analysis methods and used in classification methods. The results were compared using dimension reduction and feature selection methods.

As a result, we have reached a success rate of 72% thanks to the Neural Network algorithm.

Keywords: Machine learning algorithms, music recommendation system, classification system, using acoustic properties, audio, music, signal processing, spectral features, MFCC, CHROMA CQT

1

Giriş

Hayatın getirdiği yeni düzene ayak uydurabilmek adına teknolojik gelişmelere tanık olabilmekteyiz. Her alanı kapsayan bu teknolojik gelişmeler, beraberinde akıllı sistemleri de meydana getirmektedir. Bugünkü koşulda bir çok alanda otomasyon sistemleri hatta yapay zeka entegreli sistemler kullanılmaktadır. Yapay zeka uygulamaları sayesinde makineler, verileri analiz edip bunlardan anlamlı sonuçlar çıkartarak işlerimizi kolaylaştırmaktadır.

Bu projemizde makine öğrenmesi uygulamalarıyla müzik türü sınıflandırma üzerinde çalışma yapıldı. GTZAN veri setindeki ses dosyalarından sayısal işaret işleme teknikleri kullanılarak özellik çıkarımı yapıldı. 11 farklı makine öğrenmesi algoritması çıkartılan özellikler ile eğitildi. Makine öğrenmesi algoritmalarının sonuçları farklılık gösterse de en yüksek sonuçları veren algoritmaların başarımlarını artırmak hedeflendi. Başarımları artırmak ve özellikleri sadeleştirmek adına PCA ve ileri yönlü seçme (forward selection) yöntemleri kullanıldı. PCA sayesinde verilerde sıkıştırmaya gidilerek boyut indirgeme işlemleri gerçekleştirildi. İleri seçim yöntemi sayesinde de özelliklerin sayısı %90 azaltılmasına rağmen ilk tahminlere yakın sonuçlar elde edildi.

Bu çalışma 5. bölümde bahsedilen özellik çıkarımı yöntemleriyle detaylı bir şekilde ele alındı. Özellik çıkarımı yöntemleri sayesinde GTZAN veri setindeki 1000 ses dosyasının 276 adet özelliği çıkartıldı. Bu özellikler 7. bölümde bahsedilen makine öğrenmesi algoritmalarında eğitim amaçlı kullanıldı. Elde edilen özelliklerde iyileştirmelere gidilerek türev alma, özellik seçimi ve boyut indirgeme yöntemleri kullanıldı. Makine öğrenmesi algoritmalarının başarımlarını test etmek amacı ile model performansı değerlendirme metrikleri kullanıldı. Karmaşıklık matrisinde bulunan değerlerden doğruluk, keskinlik, duyarlılık ve F1 skorları elde edildi. Tahminlerin rassal olup olmadığını test etmek amacıyla da Cohen's Kappa katsayısından yararlanıldı. 9. bölümde ise deneylerden elde edilen sonuçlara yer verildi.

Sonuç olarak geldiğimiz noktada, veri setimiz üzerinde özellik çıkarımı, boyut

indirgeme, özellik seçimi yöntemleri kullanılarak çeşitli makine öğrenmesi algoritmaları parametre optimizasyonu ile eğitildi. Boyut indirgeme ve özellik seçimi yöntemleri, maliyeti düşürmeyi ve veri kümemizi daha az özellik ile ifade etmemizi sağlasa da modellerin başarısı üzerinde pozitif bir etki yaratmadı. Özellik çıkarımı sonrası elde edilmiş, her iki türevi de içeren veri kümesi ve Sinir Ağı algoritması ile elde ettiğimiz en yüksek başarımlı oranı olan %72'ye ulaşarak proje tamamlandı.

2 Ön İnceleme

Bu bölümde müzik setleriyle ilgili daha önce yapılmış çalışmaların literatür incelemelerine yer verilmiştir.

2002 yılında "Musical Genre Classification of Audio Signals" isimli makale George Tzanetakis tarafından yayımlandı. Toplanan veri seti, müzik türü sınıflandırması için kullanıldı ve bu veri seti tüm akademik camiaya paylaşıldı. GTZAN veri setiyle ilk yapılan çalışma olmasına rağmen %61 (Gerçek zamanlı olmayan) ve %44 (Gerçek zamanlı) başarımlar elde edildi. [1]

"Improved Music Genre Classification with Convolutional Neural Networks" makalesinde bahsedildiği üzere öncelikle GTZAN veri seti üzerinde değişiklikler yaparak daha başarılı sonuçlar elde edilmiştir. Veri setindeki 1000 adet müzik dosyası 30 saniyeden oluşmaktadır. Weibin Zhang ve ekibi bu müzik setinden her bir müzik için üç saniyelik örneklem alarak daha başarılı sonuçlara ulaşmayı başarmıştır. Veri setinin öz niteliklerini çıkarmak adına yapay sinir ağları kullanılmış ve bu sayede başarımlar max and average-pooling yöntemleriyle %87.4 olmuştur. [2]

"Evaluation of Feature Extractors and Psycho-Acoustic Transformations For Music Genre Classification" makalesinde bahsedildiği üzere GTZAN, ISMIRrhythm ve ISMIRgenge olmak üzere 3 farklı veri seti üzerinde çalışma yapılmıştır. Elde edilen çalışmalarda GTZAN veri setinden çıkarılan özelliklerle mean değerleri kullanılarak %72.85 başarıma ulaşılmasına rağmen ISMIRgenge veri setinde mean değerleri kullanılarak bu başarımlar %78.53'e kadar çıkarılmıştır. Seçilen eğitim kümesinin başarımları etkileyebileceği tekrardan gözlemlenmiştir. [3]

"Aggregate features and ADABOOST for music classification" makalesinde bahsedildiği üzere MIREX 2005 dataset kullanılarak özellik çıkarım yöntemlerine başvurulmuştur. Kullanılan makine öğrenmesi algoritmalarından AdaBoost sayesinde genel ortalamada %82.34 başarımlarına ulaşılabilmiştir. [4]

"Music Genre Classification using Machine Learning Techniques" makalesinde ise

Audio Set veri setindeki ses dosyalarını kullandığı gözlemlendi. 7 farklı müzik türü ve toplamda 40540 ses dosyasını işleyip yapay sinir ağını kullanarak eğitim sağlanmış ve spektrogram tabanlı modeller kullanarak %89.1 başarıma ulaşılmıştır. [5]

Bu bölümde projenin fizibilitesini inceleyeceğiz. Fizibilite analizi bize projede nasıl ilerleyeceğimizi gösterecek.

İlk önce teknik fizibiliteyle başlayacağız. Teknik fizibilite iki bölümden oluşuyor. İlk bölüm yazılım fizibilitesi, burda kullanacağımız uygulama geliştirme ortamları, programlama dilleri ve kütüphanelerden bahsedeceğiz. İkinci bölüm olan donanım fizibilitesinde ise geliştirme sürecinde kullanacağımız donanımları seçeceğiz.

Devamında ise projemizin ilerleyeceği süreci analiz ettiğimiz zaman fizibilitesi ve almamız gereken lisans, izinler için yasal fizibiliteyi inceleyeceğiz.

3.1 Teknik Fizibilite

Teknik fizibiliteyi yazılım ve donanım olarak 2 farklı grupta inceleyeceğiz.

3.1.1 Yazılım Fizibilitesi

Bu projede programlama dili olarak kullanmak için Python'u ve geliştirme ortamı olarak Jupyter Notebook'u seçtik. Python ve Jupyter Notebook'u seçme sebeplerimiz:

- Kullanımı kolay.
- Veri bilimi için gelişmiş kütüphaneleri var.
- Ses öznitelik çıkarımı için kullanabileceğimiz kütüphaneler var.
- Topluluk desteği fazla.

Veri bilimi kütüphaneleri olarak pandas, numpy, matplotlib, scipy, sklearn; Öznitelik çıkarımı için librosa kütüphanesini kullandık.

3.1.2 Donanım Fizibilitesi

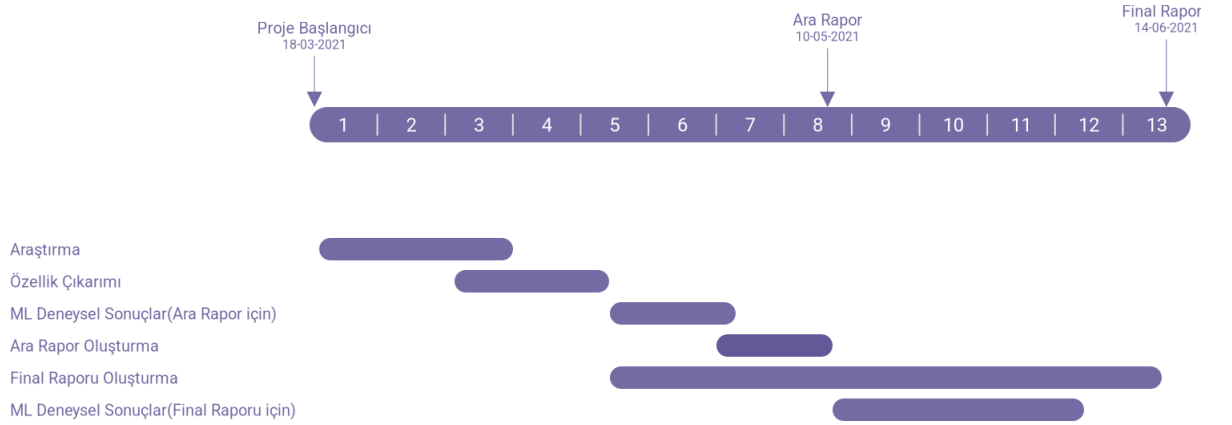
Projeyi geliştirirken ekstra bir donanım kaynağına ihtiyacımız olmadı. Kişisel bilgisayarlarımızı kullandık. Kullandığımız bilgisayarların donanımsal özellikleri:

	1. Bilgisayar	2. Bilgisayar
RAM	8 GB	16 GB
Depolama	256 GB SSD	240 GB SSD
İşlemci	i7-7700HQ	i7-7700HQ
Grafik Kartı	GeForce GTX 1050 Ti	GeForce GTX 1050
İşletim Sistemi	Windows 10 Pro 64 Bit	Windows 10 Home 64 Bit

Tablo 3.1 Donanım Bilgileri

3.2 Zaman Fizibilitesi

13 haftalık süreçte bitirilmesi planlanan projenin Gantt diyagramı şekil 3.1’de gösterilmiştir



Şekil 3.1 Gantt Çizelgesi

3.3 Yasal Fizibilite

Sadece açık kaynaklı kütüphaneler ve veri setlerini kullandığımız için ekstra bir izin almaya gerek yoktur. Kullandığımız bazı kütüphaneler ve izinleri:

numpy library BSD 3-Clause License.

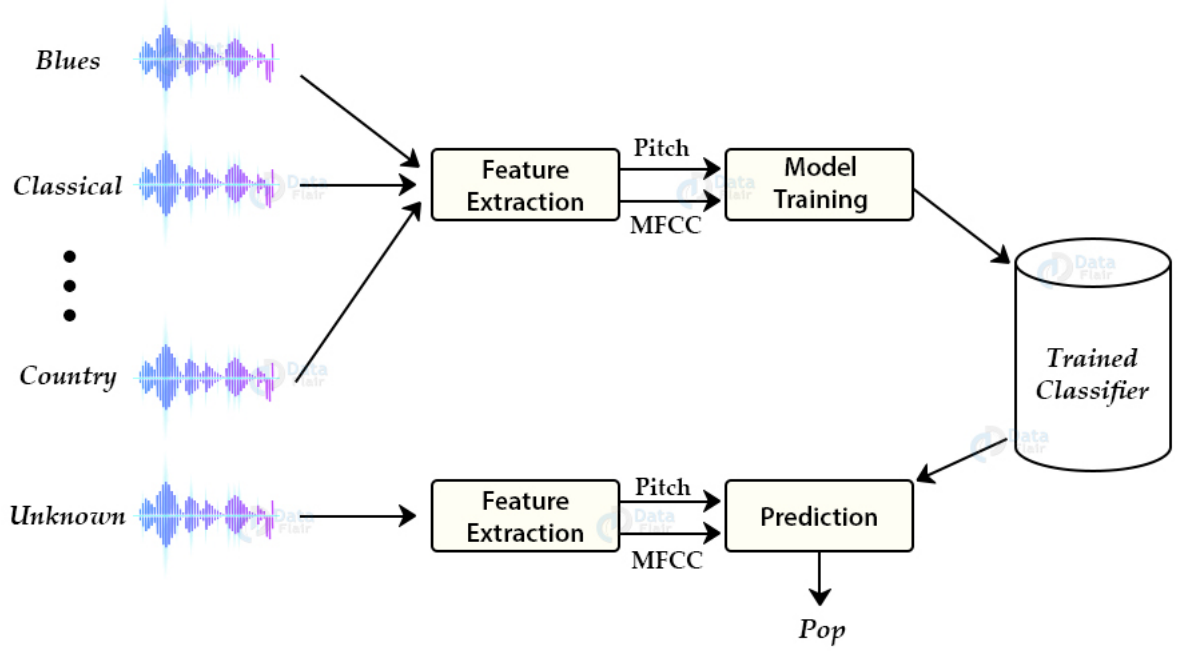
pandas library BSD 3-Clause License.

scikit-learn BSD 3-Clause License.

4 GTZAN Veri Seti

4.1 GTZAN Veri Seti Nedir?

GTZAN 2000-2001’de toplanan çeşitli müzik türlerini içerisinde barındıran bir veri setidir. Veri seti, her biri 30 saniye uzunluğunda 1000 adet ses parçasından oluşur. Her biri 100 parça ile temsil edilen 10 tür içerir. Parçaların tümü .wav formatındaki 22050 Hz Mono 16-bit ses dosyalarıdır.



Şekil 4.1 Sınıflandırma Süreci
[6]

Müzik türleri:

- blues
- klasik
- country
- disko
- hiphop
- caz
- metal
- pop
- reggae
- rock

4.2 Müzik Türü Sınıflandırma Yaklaşımları:

Bu veri setinde sınıflandırma yapmak için çeşitli yöntemler vardır. Bu yaklaşımlardan kullandıklarımız:

- Spektral Özellikler
 - Spektral Ağırlık Merkezi
 - Spektral Bant Genişliği
 - Spektral Kontrast
 - Spektral Düzlük
 - Spektral Dönüm Noktası
- Sıfır Geçiş Oranı
- Polinom Özellikleri
- Karakök Ortalama Karelerin Enerjisi
- Mel-Frekansı Kepstral Katsayıları
- Kroma Sabit-Q Dönüşümü

4.3 Özellik çıkarma

Müzik türü sınıflandırma projesinin ilk adımı, ses dosyalarından özellikleri ve bileşenleri çıkarmak olacaktır. Dilsel içeriğin tanımlanmasını ve gürültünün atılması hedeflenmektedir.

4.4 Özet:

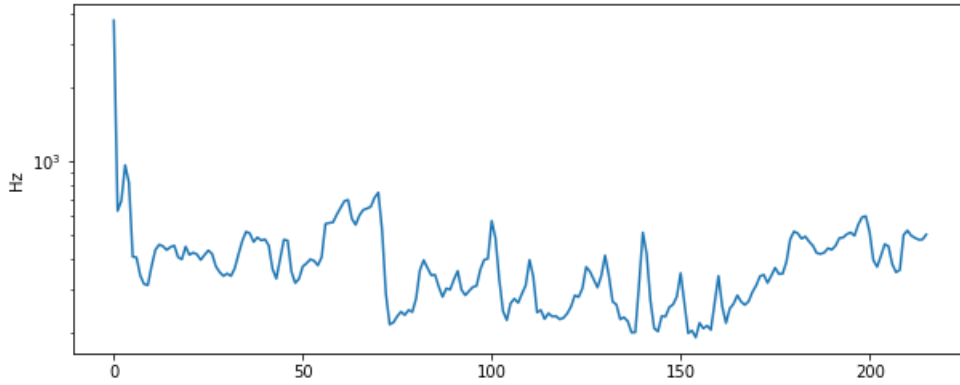
Bu müzik türü sınıflandırma projesinde, türünü tahmin etmek için ses dosyaları üzerinde bir sınıflandırıcı geliştirdik. Bu proje üzerinde GTZAN müzik türü sınıflandırma veri seti üzerinde çalışıyoruz.

5.1 Spektral Ağırlık Merkezi (Spectral Centroid)

Spektral ağırlık merkezi, sinyalin kütle merkezinin bulunduğu frekansları gösterir. Sesin parlaklığıyla güçlü bir ilişkisi bulunur. Fourier dönüşümü alınmış bir sinyalin, frekans değerlerinin ağırlıklı ortalaması ile hesaplanır. [7]

$$\frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$$

$x(n)$ ağırlıklı frekans değeri ya da büyüklük, n bulunduğumuz frekans aralığı, $f(n)$ ise aralıktaki merkez frekansı ifade eder.

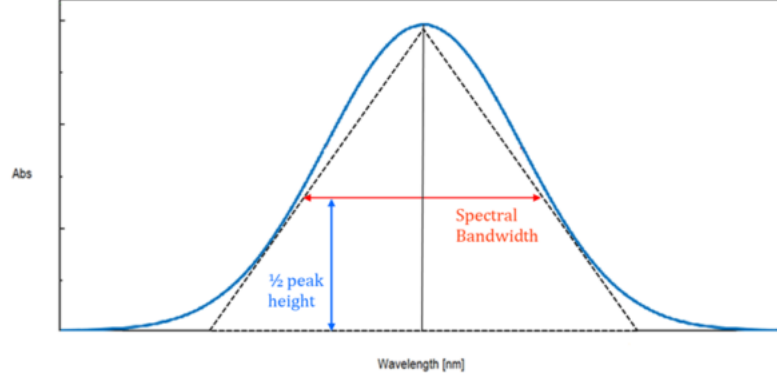


Şekil 5.1 Spektral Ağırlık Merkezi
[8]

5.2 Spektral Bant Genişliği (Spectral Bandwidth)

Ses sinyalinin dalga genişliğinin maksimum tepe noktasının yarısını tanımlar. [9]

$$\left(\sum_k S(k) (f(k) - f_c)^p \right)^{\frac{1}{p}}$$

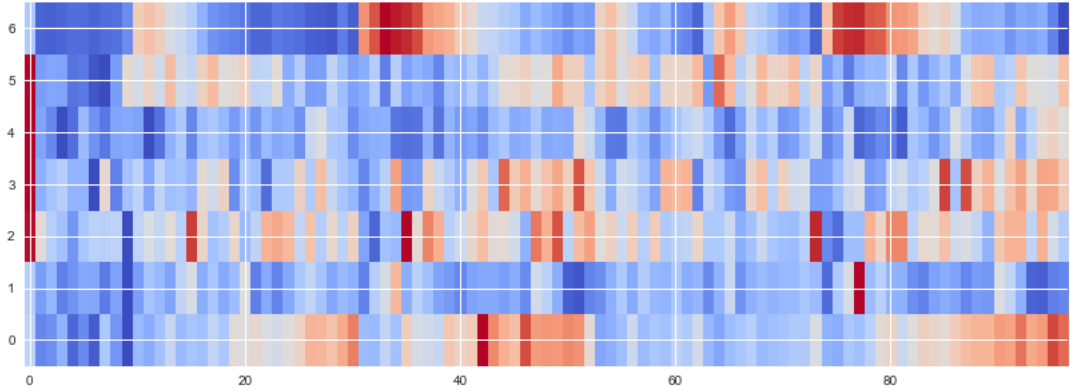


Şekil 5.2 Spektral Bant Genişliği
[10]

5.3 Spektral Kontrast (Spectral Contrast)

Spektral kontrast, spektrumda bulunan tepe ve çukur noktaları arasındaki büyüklük farkı olarak ifade edilir. [11]

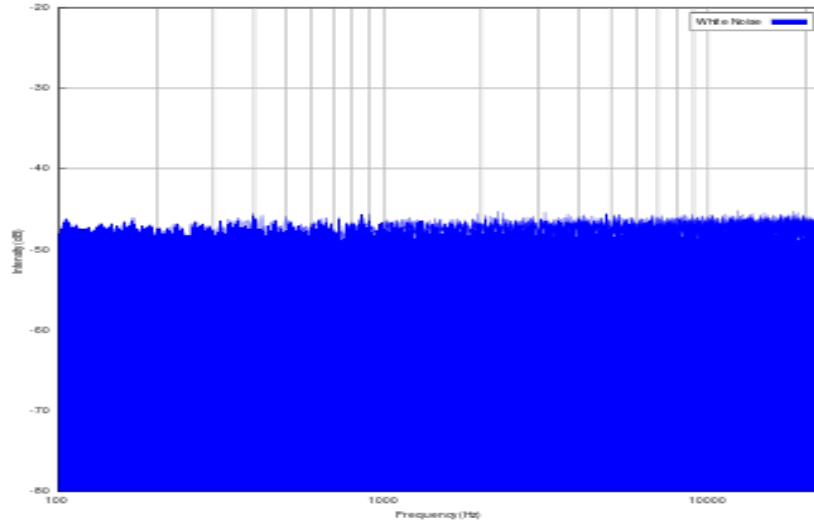
Tsinghua Üniversitesi'nin yapmış olduğu bir çalışmada spektral kontrast özelliğinin, MFCC özelliklerine kıyasla müzik türü sınıflandırmada daha üstün başarı sağladığı görülmüştür. [12]



Şekil 5.3 Spektral Kontrast
[13]

5.4 Spektral Düzlük (Spectral Flatness)

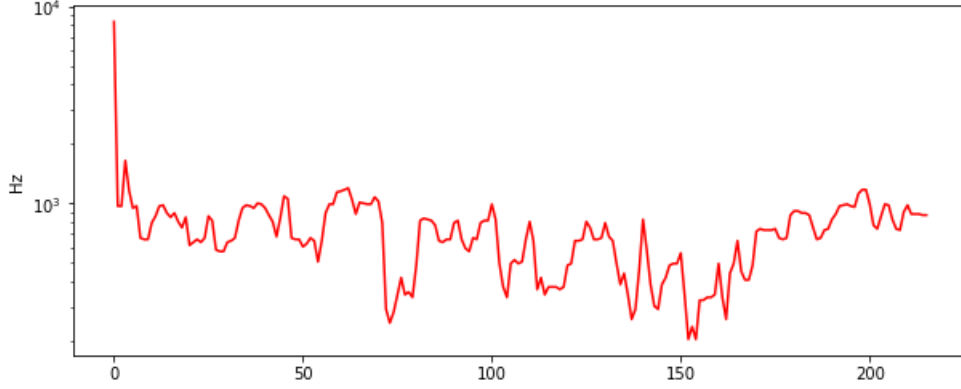
Spektral düzlük (veya tonalite katsayısı), ton benzeri olmanın aksine bir sesin gürültüye ne kadar benzediğini ölçmek için kullanılan bir ölçüdür. Yüksek bir spektral düzlük (1.0'a yakın), spektrumun beyaz gürültüye benzer olduğunu gösterir. Genellikle desibele çevrilir. [14] Güç spektrogramının geometrik ortalamasının, güç spektrogramının aritmetik ortalamasına bölünmesiyle bulunur.



Şekil 5.4 Spektral Düzlük
[15]

5.5 Spektral Dönüm Noktası (Spectral Rolloff)

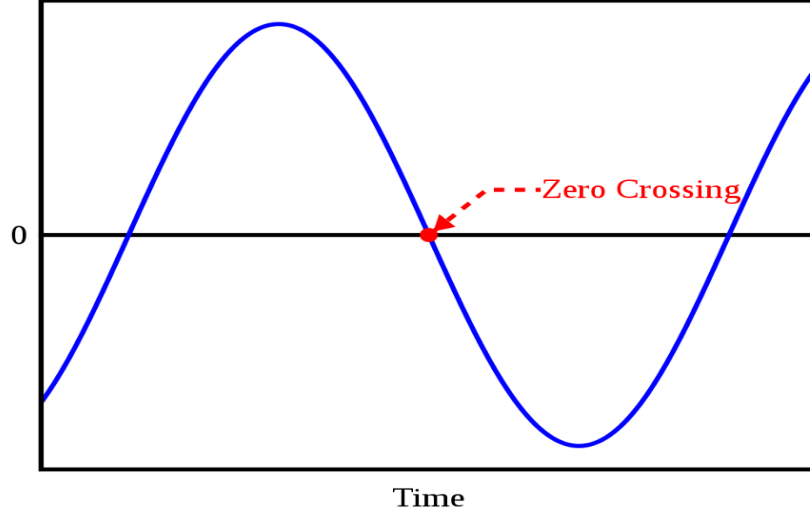
Spektral dönüm noktası, spektrumun sahip olduğu enerjinin belirli bir yüzdesinin (%85 gibi) hangi frekans değerine kadar ulaşıldığını bulmamızı sağlar. Yüzdelik dilime ulaşıldıktan sonraki frekans değeri Rolloff Frekansı olarak kullanılır. [16]



Şekil 5.5 Spektral Dönüm Noktası
[8]

5.6 Sıfır Geçiş Oranı (Zero Crossing Rate)

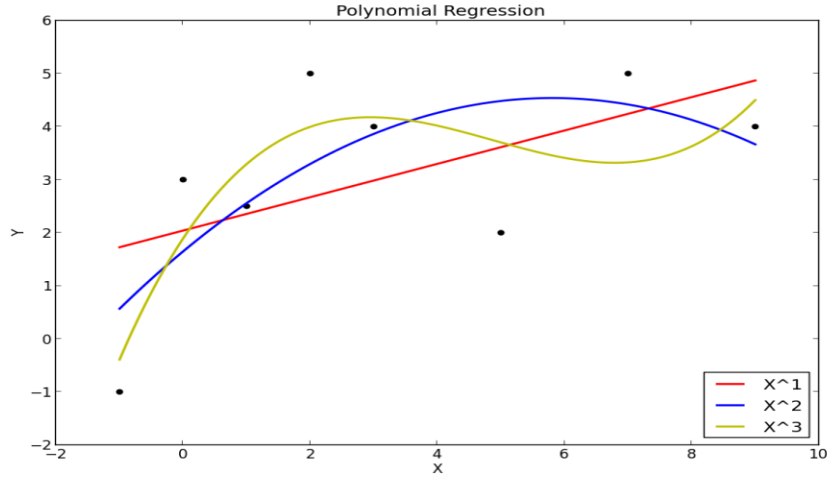
Sıfır Geçiş Oranı bir sinyalin işaret değişiminin, oranıdır. Sinyalin sıfır noktasından geçiş sayısı hesaplanarak bulunur. Konuşma tanıma ve müzik bilgilerinin alınmasında kullanılır. Vurmalı çalgıları tespit etmeyi kolaylaştırır. Hesaplaması kolaydır, gürültü tespitinde kullanılabilir. [17]



Şekil 5.6 Sıfır Geçiş Oranı
[8]

5.7 Polinom Özellikleri (Polynomial Features)

Polinom özellikleri sinyalin spektrogramına n-üslü polinom uygulanarak, oluşturulan polinomun katsayıları ile elde edilir. Doğrusal ilişki içeren veri setlerinde, problem lineer yaklaşımlarla çözülebilse de, gerçek hayatta karşılaşılan çoğu problem için lineer çözümler yetersiz kalmaktadır. Doğrusal olarak ifade edemediğimiz ilişkileri modellemek için Polinom Özellikleri kullanılabilir. [18]



Şekil 5.7 Polinom Özellikleri
[19]

5.8 Karakök Ortalama Karelerin Enerjisi (Root Mean Square Energy)

Bir sinyalin enerjisi, sinyalin toplam büyüklüğüne karşılık gelir. Bu büyüklük sinyalin ne kadar yüksek sesli olduğunu anlamamız için yeterli olmaz. Ses sinyalleri için RMSE, sinyalin ne kadar yüksek sesli olduğunu hesaplamamıza yardımcı olur. [20]

$$RMSE = \sqrt{\frac{1}{N} \sum_n |x(n)|^2}$$

5.9 Mel-Frekansı Kepstral Katsayıları (Mel-Frequency Cepstral Coefficients)

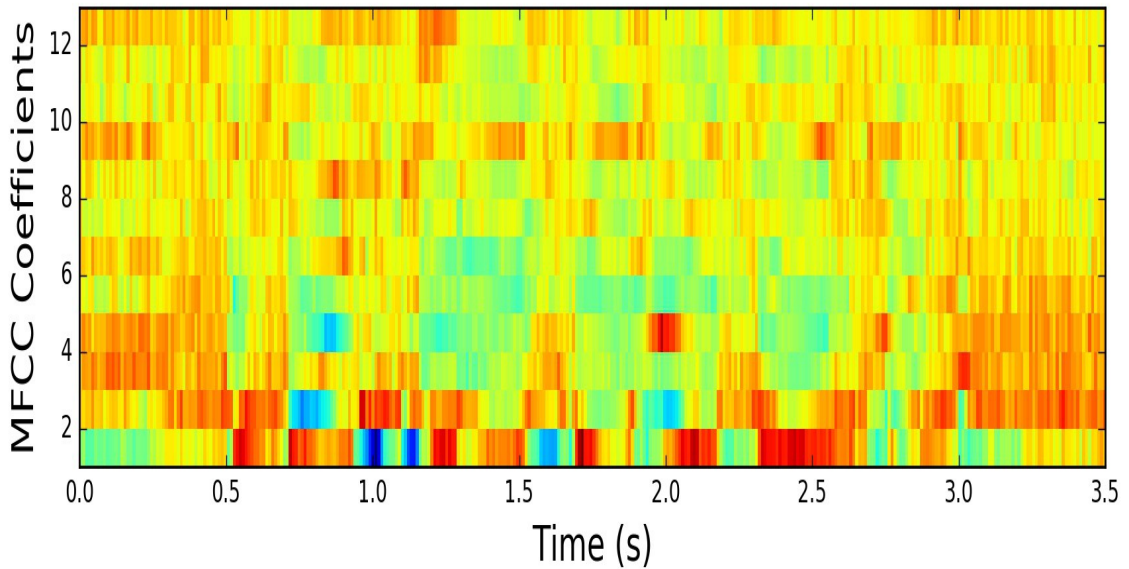
Mel frekans ölçeği, insan kulağının ses frekanslarındaki değişimi algılayışını gösteren bir ölçektir. MFCC (Mel-Frequency Cepstral Coefficients), ses sinyalinin kısa zamanlı güç spektrumunun Mel ölçeği üzerindeki ifadesidir. [21]

Frekanstan Mel ölçeğine dönüşüm formülü;

$$M = 1125 \times \ln(1 + (f \div 700))$$

$M \rightarrow$ Mel Ölçeği

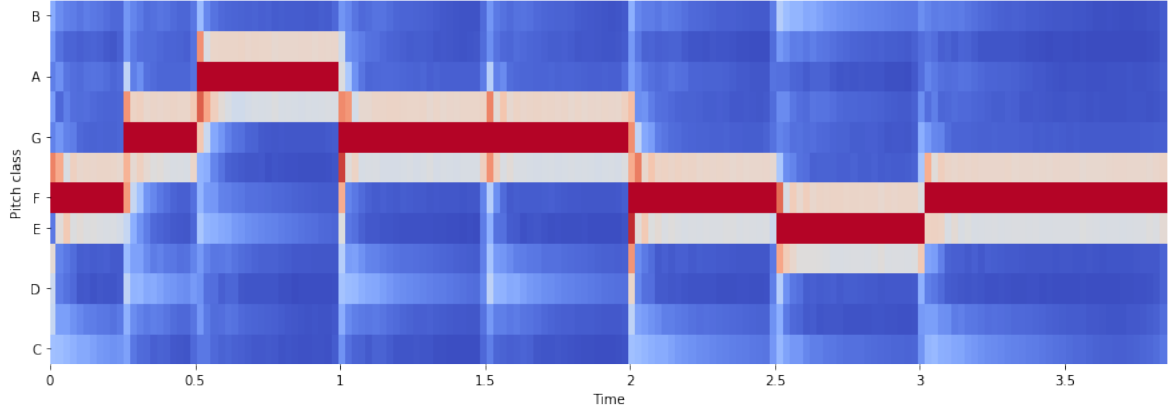
$f \rightarrow$ Frekans(Hz)



Şekil 5.8 Mel-Frekans Cepstrumu
[22]

5.10 Kroma Sabit-Q Dönüşümü (Chroma Constant-Q Transform)

Fourier dönüşümünün aksine, Kroma Sabit-Q Dönüşümü, Mel Ölçeği gibi logaritmik aralıklara bölünmüş frekansları kullanır. Chroma CQT ise batı müziğindeki 12 perdenin sinyal boyunca sahip olduğu enerjiyi 12 öznitelikli vektör olarak gösterir. [23]



Şekil 5.9 Sabit-Q Dönüşümü

[24]

6.1 Temel Bileşen Analizi (Principal Component Analysis)

Temel bileşen analizi sıkıştırma maksadıyla kullanılır ve az veriyle çok iş yapmamıza olanak sağlayan bir çeşit tekniktir. Temel amacı çok fazla veriye sahip olan veri setlerinde en yüksek varyans ile veri setini tutmaktır. Aynı zamanda boyut indirgemeyi hedefler. Yüksek adetli veri setlerinde veriler arasındaki genel özellikleri bularak boyut sayısının azaltılmasını yani verilerin sıkıştırılmasını sağlar. Özelliklerin azaltılması veri kaybına yol açsa da kazanılan zaman ve işlem kabiliyetine göre oldukça tasarruflu olabilmektedir.

PCA temelinde çok boyutlu veriyi sıkıştırarak temel özelliklere sahip az değişkenli veri haline getirir.

6.2 İleri Seçim (Forward Selection)

Veri setindeki özellik(değişken) çokluğu bize çeşitli zorlukları da beraberinde getirmektedir. Verilerin fazla oluşu güçlü donanımlara sahip olmayı gerektirir. Bu süreçte akılcı yaklaşımlar kullanılarak bu donanım maliyetinden kaçma ve eğitim sürecinde harcanan zamanı kısaltmanın çeşitli yöntemleri bulunur. Elimizdeki verilerin sayısını seçerek azaltabilirdik bu maliyetlerden kaçınma şansımız ortaya çıkar.

6.2.1 P-Value Değeri

İstatistiksel modellemenin bir çoğunda kullanıldığı gibi makine öğrenme algoritmalarında da kullanılan p-value değeri, anlamlılık düzeylerini temsil eder. Bu değer 0 ile 1 arasında modellenir. 0'a yaklaştıkça anlamlılık düzeyi artar. 0.05'ten küçük p-value değerleri anlamlandırma için kullanılabilir. İleri seçim yöntemi ile modele en fazla katkı sağlayacak değişkenin seçimi ile işe başlanır. Burada kullanılan p-value değerleri ne kadar düşük olursa o kadar anlamlı olduğu manasına

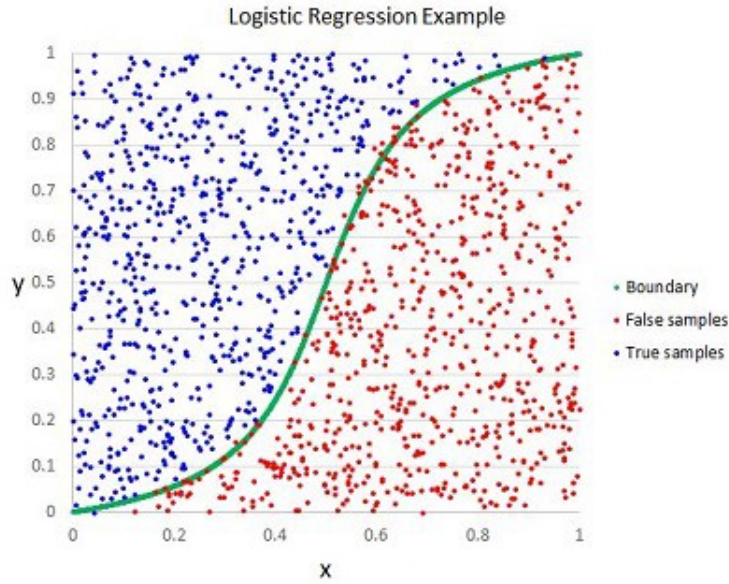
gelir.

Sonuç olarak kullanılan özelliklerin fazla olması daha anlamlı sonuçlar çıkarabileceğimiz anlamına gelmemektedir. Mantıklı seçimler yapılarak hem zamandan hem de işlem kabiliyetinden tasarruf edilebilir.

7

Makine Öğrenmesi Algoritmaları

7.1 Lojistik Regresyon (Logistic Regression)

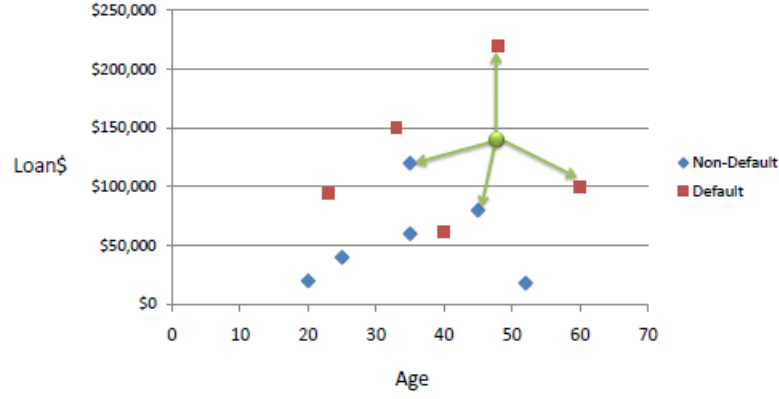


Şekil 7.1 Lojistik Regresyon
[25]

Logistic Regression (Lojistik Regresyon) değişkenlerin sayısının iki veya daha fazla olduğu durumda doğrusal bir ilişki kurma modelidir. Doğru (1) ve yanlış (0) olmak üzere mantıksal iki sonuç üretilir.

7.2 K En Yakın Komşu (K-Nearest Neighbors)

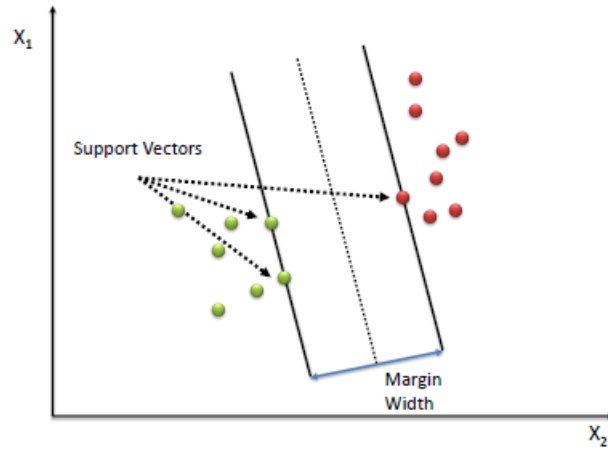
K-NN varsayımsal ilerleyen, tembel bir makine öğrenmesi algoritmasıdır. Eğitim verilerini öğrenmenin aksine veri kümesini ezberler. Bu yüzden bir tahmin yapılmak istendiği durumda, daha önceki verilerle olan yakınlık ilişkisine göre sınıflandırmayı amaçlar.



Şekil 7.2 K En Yakın Komşu
[25]

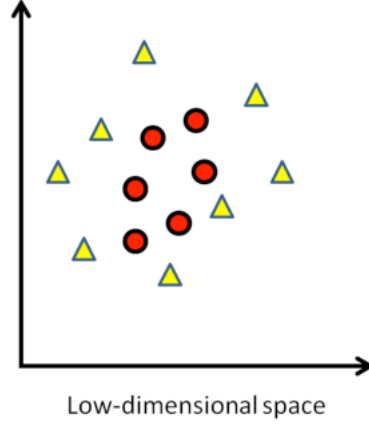
Algoritmanın çalışmasında eleman sayısı için k değeri bulunur. K'nın değerine göre en yakın K miktarında eleman alınarak gelen değer arasındaki uzaklık öklid fonksiyonu kullanılarak hesaplaması yapılır. Öklid fonksiyonuna alternatif olarak Manhattan, Minkowski ve Hamming fonksiyonları da kullanılabilir.

7.3 Destek Vektör Makinesi (Support Vector Machine)



Şekil 7.3 Destek Vektör Makinesi
[25]

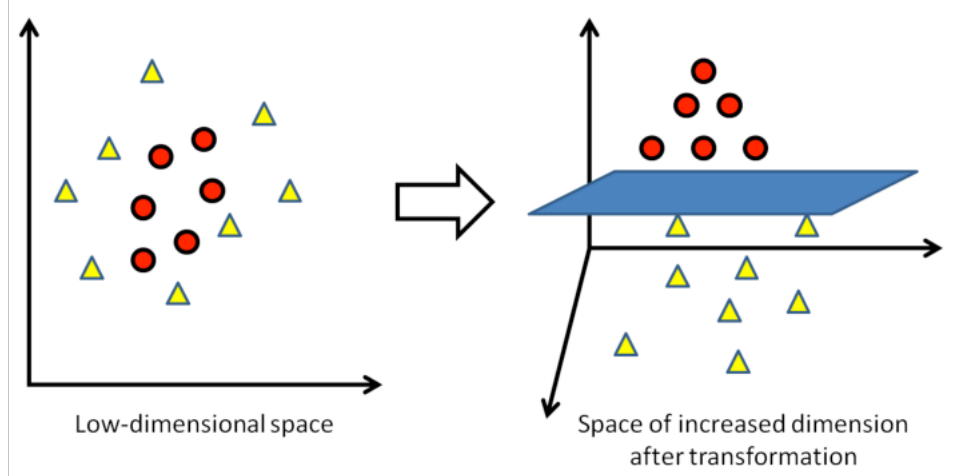
Destek vektör makinesi, doğrusal regresyon algoritmasına benzer mantıkta çalışan sınıflandırma algoritmasıdır. Her ikisi de iki sınıfı ayıran en iyi ayrıştırıcıyı bulmaya çalışır. SVM algoritması ayrılan iki sınıfın da elemanlarına en uzak yerden geçecek şekilde ayarlanmasını sağlar. Parametre almaya ihtiyaç duymaz. SVM aynı zamanda doğrusal ve doğrusal olmayan verileri sınıflandırmada da kullanılabilir.



Şekil 7.4 Çekirdek Hilesi
[25]

7.3.1 Çekirdek Hilesi (Kernel Trick)

SVM verileri lineer mantıkta sınıflandırmaya çalışır ancak bazı durumlarda 7.4 bunu yapmak imkansızdır. Bu durumdan kurtulabilmek için çekirdek hilesine (kernel trick) başvurulabilir. Farklı bir boyut oluşturabildiğimiz takdirde doğrusal olarak sınıflandırmamız mümkün olabilir. Örneğin Şekil 1' deki grafikte kırmızı noktaları biraz yukarı kaldırıp (z eksenini) 3. bir boyut oluşturabilirsek SVM ile doğrusal bir çizgi oluşturabiliriz.



Şekil 7.5 Çekirdek Hilesi 2. Boyuttan 3. Boyuta
[25]

7.4 Naive Bayes Sınıflandırıcısı (Naive Bayes)

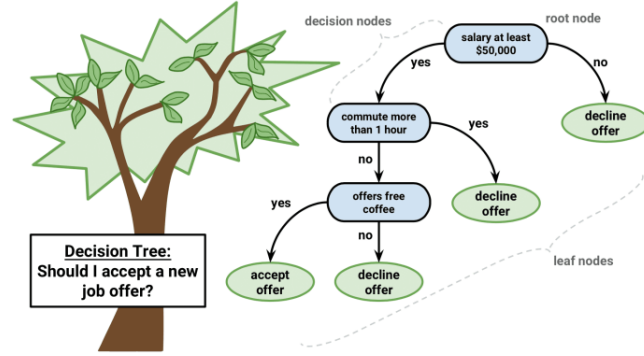
Naive Bayes, Bayes Teoremini kullanarak sınıflandırma yapan bir makine öğrenmesi algoritmasıdır. Tembel bir öğrenme algoritmasıdır. Her bir eleman için tüm olasılık değerlerini hesaplar ve olasılık değeri en yüksek olana göre sınıflandırır. Eğitim

verisi az olsa bile tahmin başarısı yüksek modeller oluşturabilir. Test kümesindeki bir değerin eğitim kümesinde bulamadığımız bir değeri varsa olasılık değeri olarak 0 verir bu yüzden tahmin yapamaz. Bu durum genellikle Sıfır Frekans (Zero Frequency) adıyla bilinir. Sıfır Frekans durumunu çözebilmek için Laplace tahmini gibi düzeltme teknikleri kullanılabilir.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|x) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

7.5 Karar Ağacı (Decision Tree)



Şekil 7.6 Karar Ağacı
[25]

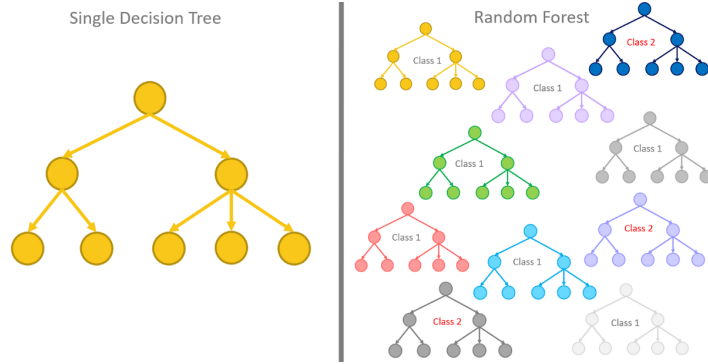
Karar ağacı (decision tree) yöntemi hem sınıflandırma (classification) hem de regresyon (regression) problemlerinde kullanılan makine öğrenme algoritmasıdır. Veri madenciliği alanında da sıkça kullanılan karar ağaçları genellikle insan seviyesinde düşünülebilecek seviyededir. Verileri anlamlandırmak ve görselleştirmek, karar ağaçları sayesinde kolayca elde edilebilir.

Karar ağacı yenilemeli (recursively) bir işlemdir. Tek bir düğüm ile başlayıp yeni sonuçların dallanmasıyla bir ağaç yapısı oluşturulur.

3 çeşit düğüm çeşidi vardır:

- Şans Düğümü (Chance Node): Daire ile gösterilir, birden çok olası yol belirtir.
- Karar Düğümü (Decision Node): Dikdörtgen ile gösterilir, bir karar verileceğini belirtir.
- Uç Düğümü (End Node): Üçgen ile gösterilir, bir sonuç belirtir.

7.6 Rassal Orman (Random Forest)

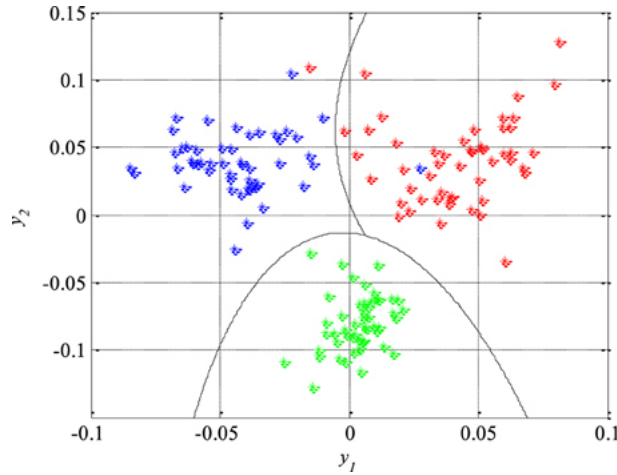


Şekil 7.7 Rassal Orman
[26]

Random Forest algoritması karar ağacı (decision tree) gibi hem classification (Sınıflandırma) hem de regresyon (regression) problemlerinde kullanılabilir. Tek karar ağacı oluşturmak yerine birden fazla karar ağacı oluşturularak elde edilir. Herhangi bir sonuç bulunmak istendiği takdirde karar ağaçlarının ortalaması alınarak bulunur.

7.7 Quadratic Diskriminant (Quadratic Discriminant)

Quadratic Classifier, iki veya daha fazla nesne ile olay sınıfının ölçümlerini ayırmak için kullanılan ikinci dereceden bir sınıflandırma yöntemidir. Doğrusal sınıflandırıcının daha kapsamlı bir versiyonudur.



Şekil 7.8 İkinci Dereceden Sınıflandırıcı
[27]

7.8 Doğrusal Diskriminant (Linear Discriminant)

Doğrusal diskriminant bir boyut indirme yöntemidir. Özniteliklerin bir doğrusal birleşimini kullanarak veriyi sınıflara ayırmayı amaçlar. Gruplardaki farklılıkları modelleyerek yüksek boyutlu bileşenleri daha düşük boyutlara indirir.

$$Z = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_d x_d$$
$$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta} \quad \text{Score Fonksiyonu}$$

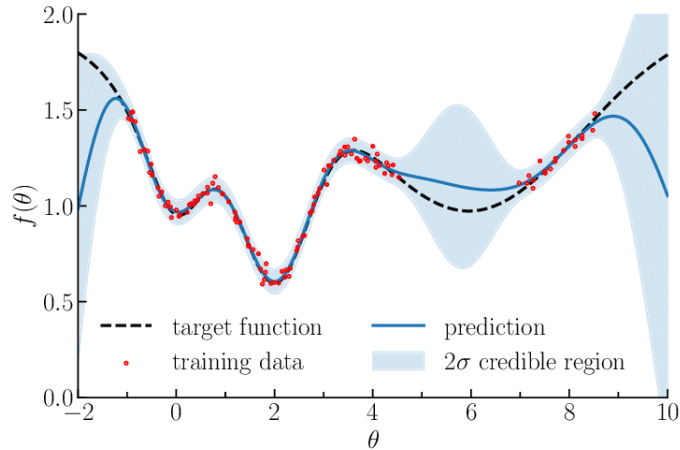
↓

$$S(\beta) = \frac{\bar{Z}_1 - \bar{Z}_2}{Z\text{'nin gruptaki varyansı}}$$

Şekil 7.9 Doğrusal Diskriminant
[28]

7.9 Gauss Süreç Regresyonu (Gaussian Process Regression)

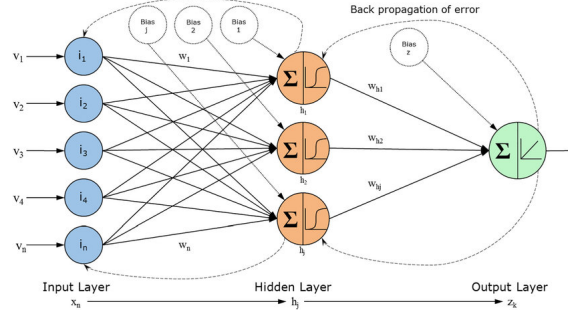
Gaussian process regression (GPR), parametrik olmayan, Bayesci bir regresyon yaklaşımıdır. GPR, küçük veri setleri üzerinde oldukça iyi çalışır. Tahminler üzerinde belirsizlik ölçümleri sağlama becerisine de sahiptir.



Şekil 7.10 Gauss Süreç Regresyonu
[29]

7.10 Sinir Ağı (Neural Network)

Yapay sinir ağı, insan beyninden bilgi işleme tekniğinde yararlanılarak geliştirilmiş bir tekniktir. İnsandaki sinir sistemlerinin çalışma mantığını taklit eder. Öğrenme, eğitim mantığıyla örnekler kullanılmasıyla gerçekleşir. İşlenen verilerin bağlantı ağırlıkları sayesinde bir yakınsama bulunana kadar tekrar tekrar denenmesi ile meydana gelir.



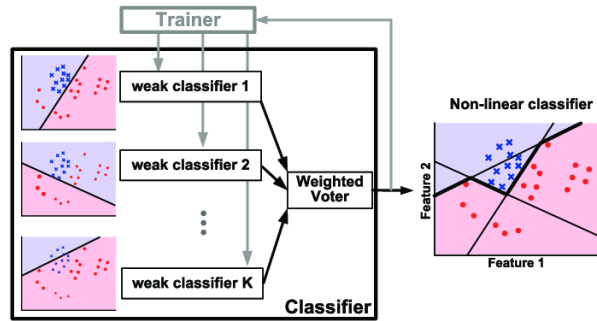
Şekil 7.11 Sinir Ağı
[30]

Beynin temel birimi nöronlar gibi yapay bir sinir ağının temel yapı taşı da basit sinyal işlemeyi gerçekleştiren bir algılayıcıdır.

Sinir ağına sahip bilgisayara önceden etiketlenmiş eğitim örneklerini analiz ettirerek bir görev yapması öğretilir. Sunulan verilerde tekrar eden kalıpları analiz ederek, yeni verileri kategorilere ayırmayı öğrenir.

7.11 Uyarlanabilir Güçlendirme (AdaBoost)

Boosting algoritmaları, her iterasyonda elde edilen zayıf öğrenicileri birleştirerek güçlü bir öğrenici elde etmeye çalışır.



Şekil 7.12 Uyarlanabilir Güçlendirme
[31]

İki değişkenin birbiriyle olan korelasyonu düşük ise tahmin oranı da düşüktür. Bunun aksine kolerasyon yüksekse tahmin oranı da yüksektir.

Bir makine öğrenmesi algoritmasını eğittikten sonra, modelin verilerimize ne kadar uygun olduğunu bulmamız gerekir. Farklı makine öğrenmesi algoritmaları için farklı model performansı değerlendirme metrikleri bulunur.

Bu projede karmaşıklık matrisi oluşturup bulduğumuz değerlerden accuracy, precision, recall ve f1 skor değerleri elde edildi. Tahminlerin rassal olup olmadığını test etmek için ise Cohen's Kappa katsayısından yararlanıldı.

8.1 Karmaşıklık Matrisi (Confusion Matrix)

Bir karmaşıklık matrisi, sonuç değerleri bilinen bir test verisi ile, herhangi bir sınıflandırma modelinin tahmin değerleri arasındaki ilişkiyi, farklı başarı metriklerine göre test etmek için kullanılır. Modelin başarısı doğru ve yanlış olarak sınıflandırılan veri sayılarının nicelikleriyle alakalıdır.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Şekil 8.1 Karmaşıklık Matrisi

- True Positive : Tahminin pozitif olup, gerçek sonucun da pozitif olduğu durumların sayısıdır.
- False Positive : Tahminin pozitif olup, gerçek sonucun negatif olduğu durumların sayısıdır.
- False Negative : Tahminin negatif olup, gerçek sonucun pozitif olduğu durumların sayısıdır.
- True Negative : Tahminin negatif olup, gerçek sonucun da negatif olduğu durumların sayısıdır.

8.2 Doğruluk (Accuracy)

Doğruluk değeri, tahmin sonucunda doğru olarak sınıflandırdığımız değerlerin toplam veri kümesine oranı ile bulunur. Tek başına yeterli derecede anlam ifade etmediği için diğer metriklerle birlikte kullanıldı.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (8.1)$$

8.3 Keskinlik (Precision)

Keskinlik ise pozitif olarak tahmin ettiğimiz değerlerden gerçekten kaçının pozitif olduğunu hesaplar.

$$Precision = \frac{TP}{TP + FP} \quad (8.2)$$

8.4 Duyarlılık (Recall)

Doğru olarak sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranıdır. [32]

$$Recall = \frac{TP}{TP + FN} \quad (8.3)$$

8.5 F1 Skor

F1 skor, precision ve recall değerlerinin harmonik ortalamasıdır ve ekstrem durumlarla başa çıkabilmek için harmonik ortalamayı kullanır. F1 skorunun kullanılmasının en temel sebebi eşit dağılmayan veri kümelerinde doğru model seçimini yapabilmektir.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (8.4)$$

8.6 Cohen's Kappa

Kappa testi iki veya daha fazla değerleyici arasındaki uyumun güvenilirliğini ölçer. Elde edilen değişkenin kategorik değişken olması sebebiyle bir parametrik olmayan istatistik türüdür.

Kappa katsayısı hesaplanırken Pr(a) ve Pr(e) olmak üzere iki farklı değişken kullanılır. Pr(a) iki değerlendirici için gözlenen uyuşmanın toplama orantısı, Pr(e) ise bu uyuşmanın şans eseri ortaya çıkma olasılığıdır. Kohen'in kappa katsayısını bulmak için kullanılan formül: [33]

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}, \quad (8.5)$$

κ	Yorum
<0	Zayıf
0.0 - 0.20	Önemsiz
0.21 - 0.40	Düşük
0.41 - 0.60	Orta
0.61 - 0.80	Önemli
0.81 - 1.00	Çoky yüksek

Tablo 8.1 Cohen's Kappa

9 Deneysel Sonuçlar

Bu bölümde, makine öğrenmesi algoritmalarını eğitirken sonuçlarımızı karşılaştırabilmek için veri setimiz üzerinde yaptığımız değişiklikleri göreceğiz. Ardından makine öğrenmesi algoritmalarının parametre değerlerini değiştirerek en optimal modeli elde etmeyi ve en yüksek skoru veren modellerin karmaşıklık matrisi, doğruluk değerleri gibi sonuçlarını inceleyeceğiz.

Makine öğrenmesi algoritmalarında en başarılı sonucu elde edebilmek için Grid-Search ve K-Fold Cross Validation yöntemlerini kullanıldı. Grid Search sayesinde modellerde en yüksek skor veren parametrelere ulaşıldı. K-Fold Cross Validation ile veri setinin tamamı test ve öğretim verisi olarak kullanılarak aşırı öğrenmenin önüne geçildi.

9.1 Veri Kümesiyle İşlemler

Özellik çıkarımı sonrası elimizde bulunan veri kümesi, aşağıda verilen öznitelikleri içeriyor:

- Spectral Centroid
- Spectral Bandwidth
- Spectral Contrast
- Spectral Flatness
- Spectral Rolloff
- Zero Crossing Rate
- Root Mean Square Energy
- MFCC

- Chroma CQT
- Poly Features

Model başarısını artırmak ve hangi veri kümesi ile en iyi sonucu verdiğini test etmek için özniteliklerin 1. ve 2. türevlerini alarak 3 farklı veri kümesi oluşturduk:

1. Türevi Alınmamış Öznitelikler (267 Öznitelik)
2. Türevi Alınmamış Öznitelikler + Özniteliklerin 1. Türevi (533 Öznitelik)
3. Türevi Alınmamış Öznitelikler + Özniteliklerin 1. Türevi + Özniteliklerin 2. Türevi (799 Öznitelik)

Makine öğrenmesi algoritmalarını eğitmek için tüm veri kümelerindeki özniteliklerin ortalama, standart sapma, medyan, mod, minimum, maksimum, basıklık ve çarpıklık değerlerini kullandık.

DeneySEL sonuçlarda kullanılan karmaşıklık matrislerindeki 0'dan 9'a kadar olan numaralar sırasıyla; blues, klasik, country, disko, hiphop, caz, metal, pop, reggae ve rock türlerini ifade ediyor.

9.2 Öznitelik Çıkarımı Sonrası Sonuçlar

İlk deneyde 3 veri kümesi üzerinde de boyut indirgeme veya öznitelik seçimi uygulanmadı. Veri kümeleri öznitelik çıkarımı yapıldıktan sonraki durumlarıyla makine öğrenmesi algoritmalarına verildi. Üç kümenin de parametre optimizasyonu yapılmış 5 Katlı Çapraz Doğrulama skorları ve en iyi skordaki parametreleri aşağıdaki tablolarda verilmiştir:

Algoritma	1. Veri Kümesi	2.Veri Kümesi	3. Veri Kümesi
Nearest Neighbors	0,567	0,586	0,589
SVM	0,676	0,692	0,7
Gaussian Process	0,584	0,58	0,591
Decision Tree	0,456	0,46	0,476
Random Forest	0,56	0,599	0,611
Neural Net	0,662	0,697	0,721
AdaBoost	0,276	0,206	0,205
Naive Bayes	0,491	0,546	0,545
QDA	0,146	0,145	0,139
Logistic Regression	0,661	0,69	0,704
Linear Discriminant	0,64	0,559	0,18

Tablo 9.1 En İyi Skorlar

Algoritma	Parametreler
Nearest Neighbors	n_neighbors= 5, p= 1, weights= uniform
SVM	C= 100, degree= 3, gamma= auto, kernel= rbf
Gaussian Process	max_iter_predict= 50
Decision Tree	criterion= entropy, max_depth= 15, splitter= best
Random Forest	criterion= entropy, n_estimators= 150
Neural Net	activation= logistic, hidden_layer_sizes= 200 max_iter= 300, solver= adam
AdaBoost	n_estimators= 50
Naive Bayes	
QDA	
Logistic Regression	C=10, max_iter= 600, solver= lbfgs
Linear Discriminant	solver= svd

Tablo 9.2 En İyi Parametreler

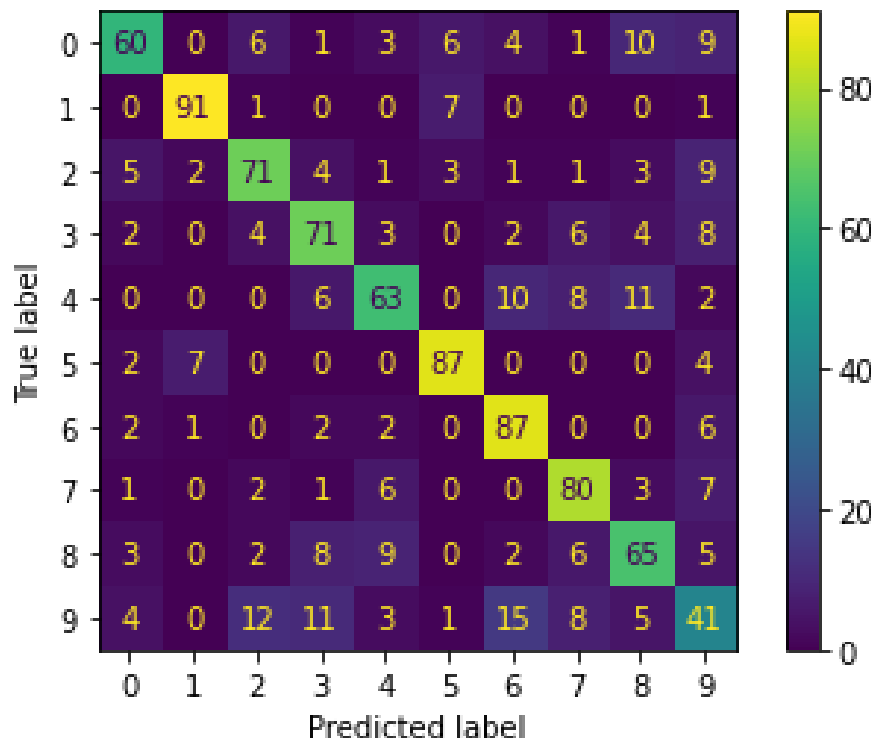
9.2.1 En İyi Sonuçlar

Makine öğrenmesi algoritmalarında her veri kümesinden en iyi sonucu veren modellerin doğruluk metrikleri karşılaştırıldı.

Veri Kümesi	Algoritma	Doğruluk	Keskinlik	Duyarlılık	F-Skoru	Cohen's Kappa
1	SVM	0,67	0,67	0,67	0,67	0,64
2	Neural Net	0,68	0,68	0,68	0,68	0,64
3	Neural Net	0,71	0,71	0,71	0,71	0,68

Tablo 9.3 En İyi Sonuçlar

Karşılaştırmalar arasında en iyi sonuç üçüncü veri kümesi kullanılarak Neural Net algoritması ile elde edildi. Neural Net algoritmasıyla elde edilen sonuçların Karmaşıklık Matrisi aşağıdaki şekilde:



Şekil 9.1 Karmaşıklık Matrisi

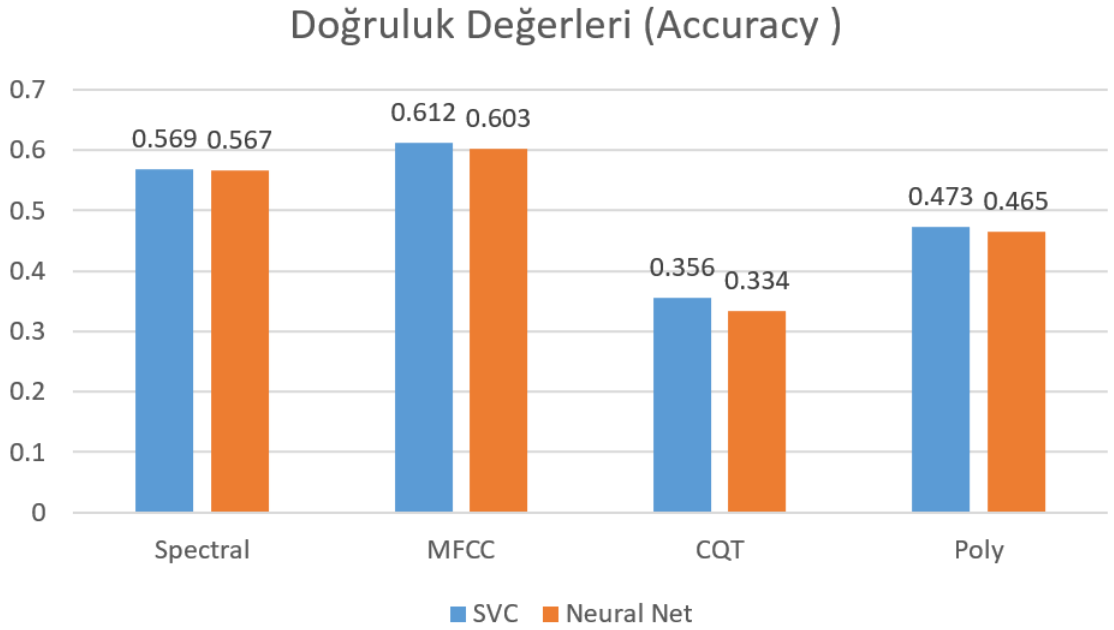
9.3 Özniteliklerin Doğruluk Skoruna Etkileri

Bu bölümde elde edilen öznitelikler türevleri kullanılmadan, sırayla SVM ve Neural Net algoritmalarına verilerek doğruluk skorları karşılaştırıldı. Kullandığımız özniteliklerin sınıflandırma üzerindeki etkilerini tespit edildi. Karşılaştırma yapmadan önce öznitelikleri 4 farklı sınıfa ayırdık:

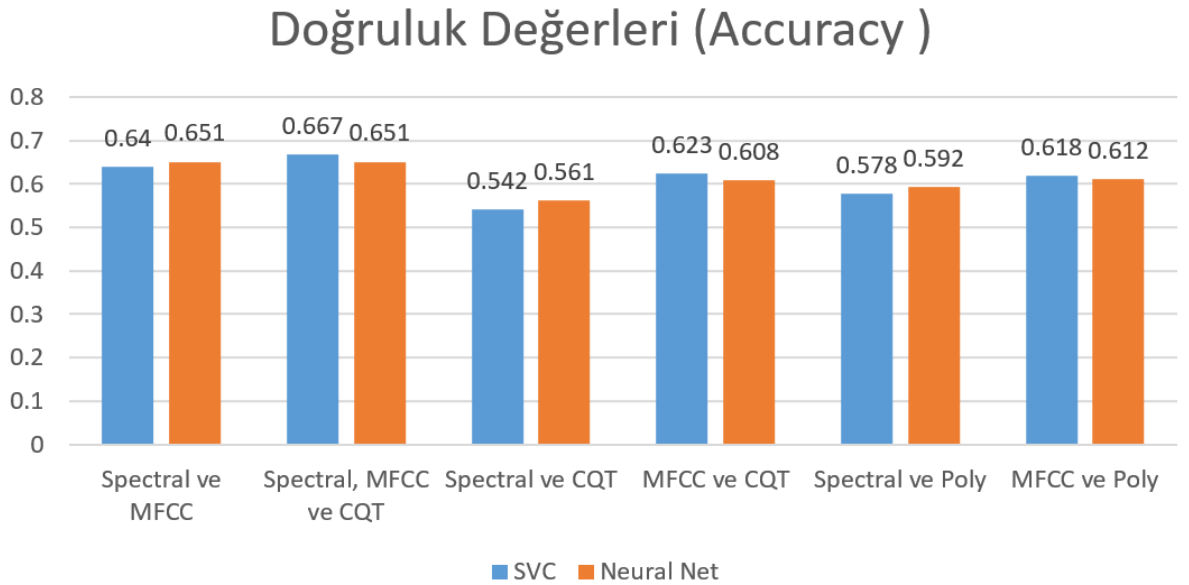
- Spektral Özellikler
- MFCC

- Chroma CQT
- Polinom Özellikleri

Yapılan karşılaştırmalar sonucunda Spektral ve MFCC özelliklerinin müzik türü tespitinde kayda değer ölçüde iyi performans verdikleri görüldü.



Şekil 9.2 Öznitelik - Doğruluk Grafiği



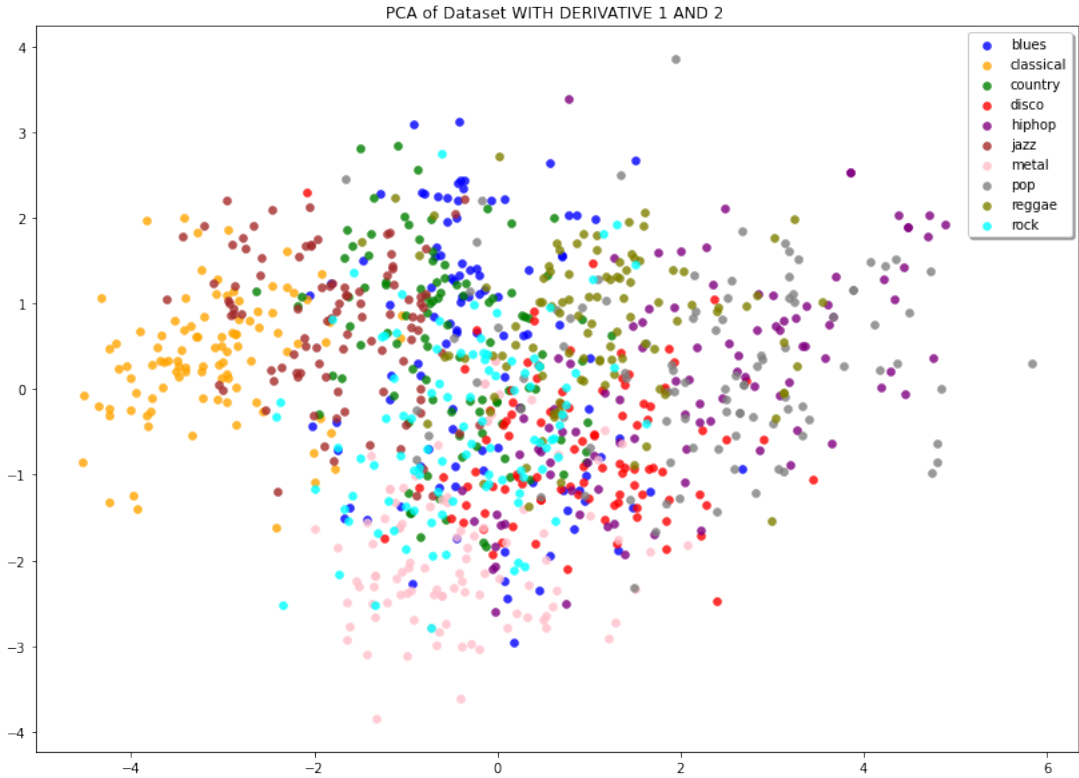
Şekil 9.3 Öznitelik - Doğruluk Grafiği

9.4 Temel Bileşen Analizi ile Sonuçlar

Bu bölümde verilerin türlerine göre 2 boyutlu bir grafikte sahip olacağı dağılım gösterildi. Ardından veri kümelerine Temel Bileşen Analizi uygulandı ve makine öğrenmesi algoritmalarından alınan sonuçlar karşılaştırıldı.

9.4.1 Dağılım Grafiği

Birinci ve ikinci türevi içeren veri kümemize Temel Bileşen Analizi uygulayarak 2 öz nitelik ile ifade edilebilmesini sağladık. Ardından verilerimizin 2 boyutlu bir grafikte nasıl dağıldığını inceledik. Bu grafik dağılım hakkında bir öngörüye sahip olmamızı sağlasa bile, öz nitelik sayımızı 2 boyuta düşürerek veri kümemizin sadece %30'unu ifade edebiliyoruz.



Şekil 9.4 Üçüncü Veri Kümesi PCA

9.4.2 Algoritma Sonuçları

Boyut indirgemeye uğramış veri kümeleri, eski veri kümelerindeki dağılımın %95'ini ifade edecek şekilde ayarlandı.

- İlk veri kümesine PCA uyguladığımızda, 75 öz nitelik ile dağılımın %95'ini ifade edebiliyoruz.

- İkinci veri kümesine PCA uyguladığımızda, 155 öznitelik ile dağılımın %95'ini ifade edebiliyoruz.
- Üçüncü veri kümesine PCA uyguladığımızda, 216 öznitelik ile dağılımın %95'ini ifade edebiliyoruz.

Lineer Diskriminant modeli diğer modellere kıyasla tüm veri kümelerinde daha iyi sonuç verdi. Temel Bileşen Analizinin boyut indirgemesi yaparken lineer yöntemler kullanması lineer sınıflandırma yapan modellerin başarısını artırdı.

PCA uygulanmış veri kümelerinden elde ettiğimiz doğruluk sonuçları ve en iyi skorların parametreleri aşağıdaki tablolarda verilmiştir:

Algoritma	1. Veri Kümesi	2. Veri Kümesi	3. Veri Kümesi
Nearest Neighbors	0,455	0,326	0,286
SVM	0,635	0,634	0,605
Gaussian Process	0,337	0,155	0,143
Decision Tree	0,384	0,415	0,408
Random Forest	0,602	0,588	0,579
Neural Net	0,605	0,622	0,616
AdaBoost	0,275	0,264	0,172
Naive Bayes	0,525	0,54	0,527
QDA	0,316	0,168	0,165
Logistic Regression	0,617	0,616	0,607
Linear Discriminant	0,656	0,682	0,681

Tablo 9.4 PCA ile En İyi Skorlar

Algoritma	Parametreler
Nearest Neighbors	n_neighbors= 2, p= 2, weights= distance
SVM	C= 5, degree= 3, gamma= scale, kernel= rbf
Gaussian Process	max_iter_predict= 50
Decision Tree	criterion= gini, max_depth= 30, splitter= best
Random Forest	criterion= gini, n_estimators= 200
Neural Net	activation= logistic, hidden_layer_sizes= 200 max_iter= 300, solver= lbfgs
AdaBoost	n_estimators= 10
Naive Bayes	
QDA	
Logistic Regression	C=10, max_iter= 100, solver= saga
Linear Discriminant	solver= svd

Tablo 9.5 PCA ile En İyi Parametreler

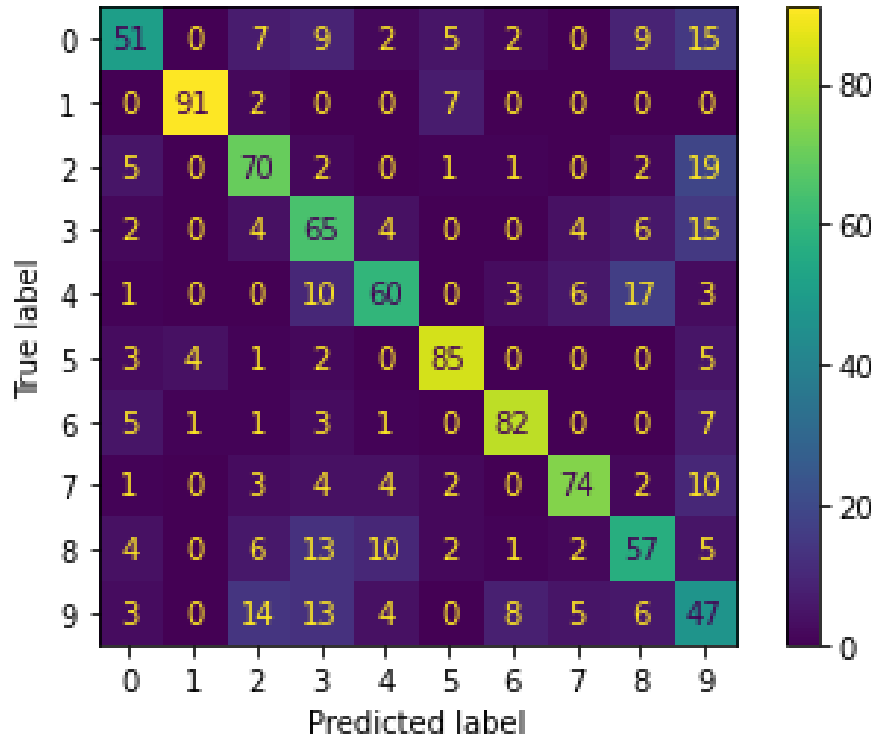
9.4.3 En İyi Sonuçlar

PCA uygulanmış veri kümeleri ile en iyi sonuç veren modellerin doğruluk metrikleri karşılaştırıldı.

Veri Kümesi	Algoritma	Doğruluk	Keskinlik	Duyarlılık	F-Skoru	Cohen's Kappa
1	<i>Linear Disc.</i>	0,65	0,67	0,65	0,66	0,61
2	<i>Linear Disc.</i>	0,68	0,69	0,68	0,68	0,64
3	<i>Linear Disc.</i>	0,68	0,69	0,68	0,68	0,64

Tablo 9.6 PCA ile En İyi Sonuçlar

Karşılaştırmalar arasında en iyi sonuç ikinci veri kümesi kullanılarak Linear Diskriminant algoritması ile elde edildi. Linear Diskriminant algoritmasıyla elde edilen sonuçların Karmaşıklık Matrisi aşağıdaki şekilde:



Şekil 9.5 Karmaşıklık Matrisi

9.5 İleri Seçim ile Sonuçlar

Projemizdeki değişkenlerin oldukça fazla olması bizim için hem zaman maliyeti hem de güçlü işlem maliyeti doğuruyordu. Onlarca değişken arsından en gerekli olanları bulup yakın tahminleme sonuçlarını elde etmeye çalıştık. Bu sayede;

- Gereğinden fazla değişkenin kullanımından kaçınarak sadeleştirmeye gittik.
- Olası aşırı öğrenme (overfitting) durumunu ortadan kaldırmaya çalıştık.
- Bazı değişkenlerin kendi arasında yüksek ilişkiye sahip (multicollinearity) olabileceğini düşünerek gereksiz değişken kullanımını azaltıp model başarısını artırmayı hedefledik.
- Değişkenleri azaltmamız sayesinde veri setini eğitim aşamasında zaman kazanmayı başardık.

9.5.1 Algoritma Sonuçları

İleri Seçim uygulanmış veri setinde, p value değeri 0,05'ten az olacak şekilde ayarlandı.

- İlk veri kümesine ileri seçim uyguladığımızda özelliklerin sayısını 267'den 27'ye indirebiliyoruz.
- İkinci veri kümesine ileri seçim uyguladığımızda özelliklerin sayısını 533'ten 59'a indirebiliyoruz.
- İlk veri kümesine ileri seçim uyguladığımızda özelliklerin sayısını 799'dan 56'ya indirebiliyoruz.

İleri seçim yöntemi sayesinde Gaussian process, AdaBoost, QDA ve Linear Discriminant algoritmalarında başarı artışı gözlemledik. Diğer algoritmalarda ise tam veri setimizdeki sonuçlara yakın değerler elde ettik. Deneyimizin sonucuna göre özellikler arasında eleme yöntemlerinin bazen daha başarılı sonuçlar üretebileceğini gözlemlemiş olduk. Bunların da yanında veri setindeki özelliklerde %90 azaltmaya gidilmesine rağmen yakın sonuçları alabildik. Bu durum bize oldukça fazla zaman kazancı sağladı.

PCA sonuçlarına yakın sonuçlar elde etmemiz bize doğru yolda olduğumuzu bir kez daha göstermiş oldu.

İleri seçim uygulanmış veri setinden elde ettiğimiz sonuçlar aşağıdaki tabloda verilmiştir.

Algoritma	1. Veri Kümesi	2.Verİ Kümesi	3. Veri Kümesi
Nearest Neighbors	0,542	0,558	0,579
SVM	0,603	0,659	0,69
Gaussian Process	0,533	0,591	0,594
Decision Tree	0,417	0,44	0,428
Random Forest	0,569	0,581	0,585
Neural Net	0,598	0,649	0,684
AdaBoost	0,25	0,223	0,235
Naive Bayes	0,453	0,484	0,505
QDA	0,456	0,445	0,438
Logistic Regression	0,694	0,642	0,691
Linear Discriminant	0,569	0,661	0,669

Tablo 9.7 İleri Yönde Seçim En İyi Skorlar

Algoritma	Parametreler
Nearest Neighbors	n_neighbors= 20, p= 2, weights= distance
SVM	C= 5, degree= 3, gamma= scale, kernel= rbf
Gaussian Process	max_iter_predict= 50
Decision Tree	criterion= entropy, max_depth= 30, splitter= best
Random Forest	criterion= gini, n_estimators= 150
Neural Net	activation= relu, hidden_layer_sizes= 100 max_iter= 800, solver= adam
AdaBoost	n_estimators= 10
Naive Bayes	
QDA	
Logistic Regression	C=10, max_iter= 100, solver= lbfgs
Linear Discriminant	solver= svd

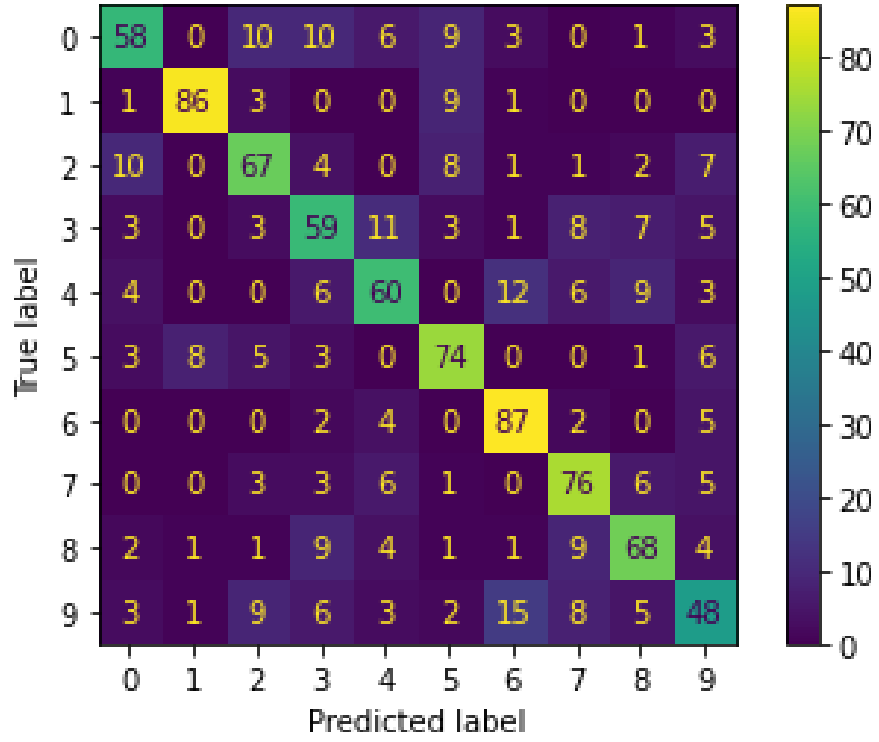
Tablo 9.8 İleri Yönde Seçim En İyi Parametreler

9.5.2 En İyi Sonuçlar

İleri seçim kullanılarak elde edilmiş veri kümeleri ile en iyi sonuç veren modellerin doğruluk metrikleri karşılaştırıldı.

Veri Kümesi	Algoritma	Doğruluk	Keskinlik	Duyarlılık	F-Skoru	Cohen's Kappa
1	SVM	0,60	0,59	0,60	0,59	0,55
2	Linear Disc.	0,66	0,66	0,66	0,65	0,62
3	Logistic Reg.	0,68	0,68	0,69	0,68	0,65

Tablo 9.9 İleri Yönde Seçim En İyi Sonuçlar



Şekil 9.6 Karmaşıklık Matrisi

10

Sonuç

Bu projede sahip olduğumuz müzik verisi setiyle makine öğrenmesi algoritmalarını eğitip, bir müziği on farklı sınıfa göre en başarılı şekilde sınıflandırabilecek algoritmayı bulmayı hedefledik.

Başlangıçta GTZAN veri setindeki ses dosyalarını alıp spektral bant genişliği, MFCC, Chroma CQT gibi ses verilerini sınıflandırmamıza yarayacak öznitelikleri çıkardık. Modellerin başarısı üzerindeki etkilerini incelemek için çıkardığımız özniteliklerin türevlerini de ayrı veri kümelerinde tuttuk. Modellerimizi eğitmek için ses verilerindeki öznitelikleri mod, medyan, standart sapma gibi matematik fonksiyonlarıyla daha anlamlı verilere dönüştürdük.

Veri kümelerimizi oluşturduktan sonra, 11 farklı makine öğrenmesi algoritmasını verilerimizle eğittik. Algoritmaların aldığı parametreleri optimize etmek amacıyla Grid Search'ten yararlandık. Bu sayede bize en yüksek skoru veren parametreleri bulduk. Verilerimizi test ve öğretim verileri olarak ikiye bölmek yerine Cross Validation kullanarak, skor hesaplarırken aşırı uyum probleminin önüne geçtik.

Parametre optimizasyonu ile birden çok makine öğrenmesi algoritmasını eğitmek maliyetli olduğu için Temel Bileşen Analizi yöntemini uyguladık. Öznitelik sayılarının düşmesiyle birlikte algoritmalar daha hızlı öğrenmeye başlasa bile bu durum başarımlar olarak önemli bir artış sağlamadı. Temel Bileşen Analizi boyut indirgeme yaparken lineer bir metod kullandığı için, Lineer Diskriminant makine öğrenmesi algoritmasının başarı oranları artış gösterdi. Öznitelik sayımızı neredeyse %75 oranında azaltarak önceki başarı değerlerine yakın sonuçlar elde ettik.

Ardından ileri yönlü özellik seçimi yöntemini kullanarak veriler üzerinde en yüksek anlama sahip öznitelikleri seçerek gereksiz özelliklerden kurtulunması amaçlandı. Özellik seçimi sonrası veri kümelerinin boyutları %90 oranında azaldı. Temel Bileşen Analizi sonucunda aldığımız başarı değerlerine yakın sonuçlar elde edildi.

Sonuç olarak, boyut indirgeme ve özellik seçimi yöntemlerinin kullanılmasından

bağımsız olarak 1 ve 2. türevlerin kullanıldığı veri kümeleri daha yüksek başarımla elde etti. İleri yönlü özellik seçimi kullandığımız deneylerde başarıda bir artış gözlenmedi. Özellik çıkarımı sonrası boyut indirgeme veya özellik seçimi yöntemleriyle değiştirilmeyen, ilk deneydeki 1. ve 2. türevlerin bulunduğu veri kümesi, Sinir Ağı algoritmasını kullanarak %72 ile en yüksek başarıyı elde etti. Özellik seçimi ile başarıda artışı sağlamak için çalışmamızı devam ettireceğiz.

- [1] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [2] W. Zhang, W. Lei, X. Xu, and X. Xing, "Improved music genre classification with convolutional neural networks.," in *Interspeech*, 2016, pp. 3304–3308.
- [3] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in *ISMIR*, 2005, pp. 34–41.
- [4] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and a da b oost for music classification," *Machine learning*, vol. 65, no. 2-3, pp. 473–484, 2006.
- [5] H. Bahuleyan, "Music genre classification using machine learning techniques," *arXiv preprint arXiv:1804.01149*, 2018.
- [6] *Python project - music genre classification*, Mar. 2021. [Online]. Available: <https://data-flair.training/blogs/python-project-music-genre-classification/>.
- [7] *Librosa.feature.spectral_centroid*. [Online]. Available: https://librosa.org/doc/0.7.2/generated/librosa.feature.spectral_centroid.html.
- [8] A. Durmaz, *Sesin öznitelik çıkarımı-librosa*, Aug. 2019. [Online]. Available: <https://medium.com/datarunner/librosa-9729c09ecf7a>.
- [9] *Librosa.feature.spectral_bandwidth*. [Online]. Available: https://librosa.org/doc/0.7.2/generated/librosa.feature.spectral_bandwidth.html.
- [10] *Theory of uv-visible spectroscopy (the basics): Jasco*, Mar. 2021. [Online]. Available: <https://jascoinc.com/learning-center/theory/spectroscopy/uv-vis-spectroscopy/>.
- [11] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proceedings. IEEE International Conference on Multimedia and Expo*, IEEE, vol. 1, 2002, pp. 113–116.
- [12] J. Yang, F.-L. Luo, and A. Nehorai, "Spectral contrast enhancement: Algorithms and comparisons," *Speech Communication*, vol. 39, no. 1-2, pp. 33–46, 2003.
- [13] *Spectral features*. [Online]. Available: https://musicinformationretrieval.com/spectral_features.html.
- [14] S. Dubnov, "Generalization of spectral flatness measure for non-gaussian linear processes," *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 698–701, 2004.

- [15] *Spectral flatness*, Feb. 2021. [Online]. Available: https://en.wikipedia.org/wiki/Spectral_flatness.
- [16] *Librosa.feature.spectral_rolloff*. [Online]. Available: https://librosa.org/doc/0.7.2/generated/librosa.feature.spectral_rolloff.html.
- [17] *Librosa.feature.zero_crossing_rate*. [Online]. Available: https://librosa.org/doc/0.7.2/generated/librosa.feature.zero_crossing_rate.html.
- [18] *Librosa.feature.poly_features*. [Online]. Available: https://librosa.org/doc/0.7.2/generated/librosa.feature.poly_features.html.
- [19] *Dauphine 203 - machine learning competition*. [Online]. Available: <https://www.kaggle.com/c/dauphine-203-machine-learning-competition-2019/discussion/84814>.
- [20] V. Bilimcisi. (2021). "Mse, rmse, mae, mape ve diğer metrikler," [Online]. Available: <https://veribilimcisi.com/2017/07/14/mse-rmse-mae-mape-metrikleri-nedir/> (visited on 05/08/2021).
- [21] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "Hmm-based audio keyword generation," in *Pacific-Rim Conference on Multimedia*, Springer, 2004, pp. 566–574.
- [22] P. Nair, *The dummy's guide to mfcc*, Jul. 2018. [Online]. Available: <https://medium.com/prathena/the-dummys-guide-to-mfcc-aceab2450fd>.
- [23] *Librosa.feature.chroma_cqt*. [Online]. Available: https://librosa.org/doc/0.7.2/generated/librosa.feature.chroma_cqt.html.
- [24] *Constant-q transform*. [Online]. Available: <https://musicinformationretrieval.com/chroma.html>.
- [25] E. Hatipoglu, *Machine learning - classification - logistic regression - part 8*, Jul. 2018. [Online]. Available: <https://medium.com/@ekrem.hatipoglu/machine-learning-classification-logistic-regression-part-8-b77d2a61aae1>.
- [26] R. Silipo, *From a single decision tree to a random forest*, Oct. 2019. [Online]. Available: <https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147>.
- [27] D. Gajic. (2021). "Piecewise quadratic classifier," [Online]. Available: https://www.researchgate.net/figure/Piecewise-quadratic-classifier-which-separates-epileptic-in-red-from-non-epileptic_fig4_274728281.
- [28] *04.06. linear discriminant analysis (lda - lineer diskriminant analizi)*, Nov. 2019. [Online]. Available: https://erdincuzun.com/makine_ogrenmesi/linear-discriminant-analysis-lda-lineer-diskriminant-analizi/.
- [29] F. Leclercq. (2021). "Illustration of gaussian process," [Online]. Available: https://www.researchgate.net/figure/Illustration-of-Gaussian-process-regression-in-one-dimension-for-the-target-test_fig1_327613136.

- [30] C. BY. (2021). "Workflow diagram of the artificial neural network algorithm," [Online]. Available: https://www.researchgate.net/figure/Workflow-diagram-of-the-artificial-neural-network-algorithm-developed-by-Lancashire-et_fig3_323312622.
- [31] Z. Wang. (2021). "Illustration of adaboost algorithm for creating a strong classifier based on multiple weak linear classifiers.," [Online]. Available: https://www.researchgate.net/figure/Illustration-of-AdaBoost-algorithm-for-creating-a-strong-classifier-based-on-multiple_fig9_288699540.
- [32] C. Coşkun and A. Baykal, "Veri madenciliğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması," *Akademik Bilişim*, vol. 2011, pp. 1–8, 2011.
- [33] S. Kiliç, "Kappa test," *Psychiatry and Behavioral Sciences*, vol. 5, no. 3, p. 142, 2015.

BİRİNCİ ÜYE

İsim-Soyisim: Engin Deniz ÇAĞLAR
Doğum Tarihi ve Yeri: 09.07.1999, Ordu
E-mail: 11117080@std.yildiz.edu.tr
Telefon: 0507 658 64 14
Staj Tecrübeleri: -

İKİNCİ ÜYE

İsim-Soyisim: Ahmet YAMAN
Doğum Tarihi ve Yeri: 17.06.1997, Isparta
E-mail: 11118607@std.yildiz.edu.tr
Telefon: 0537 654 33 00
Staj Tecrübeleri: -

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows İşletim Sistemi, Python
Gerekli RAM: 4GB
Gerekli Disk: 5GB