

**FEN BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**  
**BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ PROGRAMI**

**GÖZETİMLİ MAKİNE ÖĞRENMESİ YOLUYLA TÜRE**  
**GÖRE METİNDEN SES SENTEZLEME**

**YÜKSEK LİSANS TEZİ**

**Mehmet Ali KUTLUGÜN**

**Danışmanı: Yrd. Doç. Dr. Yahya ŞİRİN**

**İSTANBUL**

Aralık 2017

Her hakkı saklıdır.

**FEN BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİLİĞİ ANA BİLİM DALI**  
**BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ PROGRAMI**

**GÖZETİMLİ MAKİNE ÖĞRENMESİ YOLUYLA TÜRE  
GÖRE METİNDEN SES SENTEZLEME**

**YÜKSEK LİSANS TEZİ**

**Mehmet Ali KUTLUGÜN**

**Danışmanı: Yrd. Doç. Dr. Yahya ŞİRİN**

**İSTANBUL**

Aralık 2017

Her hakkı saklıdır.

## ONAY SAYFASI

**Mehmet Ali KUTLUGÜN** tarafından hazırlanan “**Gözetimli Makine Öğrenmesi Yoluyla Türe göre Metinden Ses Sentezleme**” adlı çalışma aşağıdaki jüri üyeleri tarafından BİLGİSAYAR MÜHENDİLİĞİ ANA BİLİM DALI “BİLGİSAYAR MÜHENDİSLİĞİ” programında “YÜKSEK LİSANS TEZİ” olarak kabul edilip onaylanmıştır.

**Başkan**

**Doç. Dr. Fatih KOÇAN**



**Danışman**

**Yrd. Doç. Dr. Yahya ŞİRİN**



**Üye**

**Yrd. Doç. Dr. Mehtap YALÇINKAYA**



## BEYAN

Bu çalışma İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü BİLGİSAYAR MÜHENDİLİĞİ ANA BİLİM DALI BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ'ndeki öğrenciliğim döneminde hazırlanmış olan YÜKSEK LİSANS TEZİ tarafımdan yapılmış ve kaleme alınmış tamamen özgün bir çalışma olup bu çalışmamın başından sonuna kadar bilimsel ahlak kurallarına uydum. Bu çalışmam süresince elde etmediğim ve tezimde/raporumda kullanmış olduğum bütün bilgiler ve yorumlar için atıf yaptığımı ve kaynak gösterdiğimi, patent ve telif haklarını ihlal edici bir davranışta bulunmadığımı beyan ederim.

İmza



**Mehmet Ali KUTLUGÜN**

## TEŐEKKÜR

Tez alıŐmamın gerekleŐmesi aŐamasında beni yÖnlendiren ve tecrübelerini paylaŐan tez danıŐmanı hocam Sayın Yrd. Do. Dr. Yahya ŐİRİN'e teŐekkür ederim.

Ayrıca kaynaklarından faydalandığım tüm akademik yayın yazarlarına katkılarından Ötörü teŐekkür ederim.

# İÇİNDEKİLER

<b>1</b>	<b>GİRİŞ</b> .....	<b>1</b>
1.1	Tezin Amacı .....	2
1.2	Tezin Kapsamı.....	3
1.3	Ses Sentezleme Üzerine Yapılan Araştırmalar .....	3
1.3.1	Mekanik Ses Sentezleme Sistemleri .....	3
1.3.2	Elektrikli Ses Sentezleme Sistemleri .....	4
1.3.3	Türkçe Ses Sentezleme Sistemleri .....	5
1.4	Varsayımlar .....	7
<b>2</b>	<b>SES SENTEZLEME</b> .....	<b>8</b>
2.1	Genel Bilgiler .....	8
2.1.1	Sesin Yapısı.....	8
2.1.2	Konuşma ve Özellikleri.....	9
2.2	Ses Tanıma .....	10
2.2.1	Ses Tanıma Süreci.....	11
2.2.2	Ses Sentezleme ile Neler Yapılabilir?.....	12
2.2.3	Ses Tanımının Kullanım Alanları ve Kısıtları.....	12
2.3	Metinden Konuşma Sentezleme.....	14
2.3.1	Metinden Konuşma Sentezleme Nedir?.....	14
2.3.2	MKS Kullanım Alanları ve Yararları.....	14
2.3.3	MKS Süreci ve Yöntemleri .....	15
2.3.4	MKS’de Karşılaşılan Problemler .....	16
2.3.5	MKS Sisteminde Doğallık ve Doğallığa Etki Eden Faktörler .....	18

2.3.6	Bazı Önemli MKS Uygulamaları.....	19
2.4	Türkçe Metinden Konuşma Sentezleme Sistemleri .....	21
2.4.1	Metin Önışleme Aşaması .....	23
2.4.2	Metnin Hecelere Ayrılması Aşaması .....	25
2.4.3	Ses Veri Tabanının Oluşturulması Aşaması .....	26
2.4.4	Seslerin Birleştirilmesi ve Seslendirme Aşaması.....	27
<b>3</b>	<b>MAKİNE ÖĞRENMESİ .....</b>	<b>30</b>
3.1	Makine Öğrenmesi Nedir? .....	30
3.1.1	Gözetimsiz Öğrenme.....	31
3.1.2	Gözetimli Öğrenme.....	32
3.2	Sınıflandırma.....	33
3.2.1	Bazı Önemli Kavramlar .....	33
3.3	Metinler Üzerinde İşlemler .....	35
3.3.1	Metin Önışleme.....	35
3.3.2	Terim Sayma Modeli.....	36
3.3.3	Vektör Uzayı Modeli.....	36
3.3.4	Gereksiz Verilerin Temizlenmesi .....	37
3.3.5	Kelime Köklerinin Tespiti .....	37
3.3.6	Özellik Belirleme .....	38
3.3.7	N-Gram Modeli.....	38
3.3.8	Terim Frekansları .....	39
3.3.9	Terim Ağırlığı .....	39
3.3.10	Boyut Azaltma .....	40
3.3.11	Doküman Frekansı (Document Frequency).....	41
3.3.12	Performans Ölçümü .....	41
3.4	Bazı Önemli Makine Öğrenmesi Algoritmaları .....	42
3.4.1	K-Ortalama (Means) Algoritması .....	42

3.4.2	K - En Yakın Komşu Algoritması.....	43
3.4.3	Karar Ağaçları (Decision Trees : DTs) .....	44
3.4.4	Naive Bayes Sınıflandırıcı .....	45
3.4.5	Yapay Sinir Ağları .....	46
3.4.6	Destek Vektör Makinesi (Support Vector Machine : SVM).....	47
<b>4</b>	<b>DENEYSEL ÇALIŞMA.....</b>	<b>50</b>
4.1	Bir Veri Kümesi Üzerinde Bazı Makine Öğrenmesi Algoritmalarının Karşılaştırılması.....	50
4.1.1	WEKA.....	50
4.1.2	WEKA Dosya Yapısı .....	51
4.1.3	Deneysel Çalışma.....	52
4.1.4	K - En Yakın Komşu Algoritması.....	52
4.1.5	Karar Ağaçları Algoritması.....	53
4.1.6	Naive Bayes Sınıflandırıcı .....	54
4.1.7	Destek Vektör Makineleri (Support Vector Machines) .....	55
4.1.8	Deney Sonuçlarının Değerlendirilmesi .....	56
<b>5</b>	<b>UYGULAMA GERÇEKLEME .....</b>	<b>57</b>
5.1	Uygulamanın Amacı .....	57
5.2	Uygulama Süreci .....	57
5.3	Uygulamanın Gerçekleştirimi .....	58
5.3.1	Gereksiz Kelimelerin Temizlenmesi.....	58
5.3.2	Kelime Köklerinin Tespiti .....	59
5.3.3	Doküman-Terim Matrisinin Elde Edilmesi.....	59
5.3.4	Boyut Azaltma .....	60
5.3.5	Benzerliklerin Bulunması ve Başarım Oranının Hesaplanması.....	60
5.3.6	Sınıflandırma Başarımının Değerlendirilmesi .....	61
5.3.7	Ses Dosyalarının Elde Edilmesi.....	62





## KISALTMALAR

**API** : Application Programming Interface : Uygulama Programlama Arayüzü

**ARFF** : Attribute Relationship File Format

**CPS** : Cycle Per Second : Saniyedeki çevrim sayısı

**CS** : Chi-Square : Ki-Kare

**dB** : Desibel

**DF** : Document Frequency : Doküman Frekansı

**DN** : Doğru Negatif

**DP** : Doğru Pozitif

**DTs** : Decision Trees : Karar Ağaçları

**FN** : False Negative : Yanlış Negatif

**FP** : False Positive : Yanlış Pozitif

**Hz** : Hertz

**IDF** : Inverse Document Frequency : Ters Doküman Frekansı

**IG** : Information Gain : Bilgi Kazanımı

**KHz** : Kilo Hertz

**k-NN** : k Nearest Neighbour – K En Yakın Komşu

**KTS** : Konuşma Tanıma Sistemi

**MI** : Mutual Information : Karşılıklı Bilgi

**MKS** : Metinden Konuşma Sentezleme

**OLA** : Örtüşme Ekleme Algoritması

**PAT** : Parametric Artificial Talker : Parametrik Yapay Konuşmacı

**SAPI** : Microsoft System Speech Application : Microsoft Sistem Konuşma Uygulaması

**SDK** : Software Development Kit : Yazılım Geliştirme Kiti

**SOLA** :Eşzamanlı Örtüşme-Ekleme Algoritması

**SVM** : Support Vector Machine : Destek Vektör Makinesi

**TA** : Terim Ağırlıklandırma

**TASA** : Turkish Automatic Spelling Algorithm : Türkçe Otomatik Heceleme Algoritması

**TD-PSOLA** : Zaman-Alan Senkronize Ton Örtüşme ve Ekleme Algoritması

**TF**: Term Frequency : Terim Sıklığı

**TMKS** : Türkçe Metinden Konuşma Sentezleme

**TN** : True Negative : Doğru Negatif

**TP** : True Positive : Doğru Pozitif

**TTS** : Text-to-Speech – Metinden Ses

**WAV** : Waveform Audio File : Dalga Biçimi Ses Dosyası

**WEKA** : Waikato Environment for Knowledge Analysis

**WSOLA** : Dalga Şekilli- Benzerlik Tabanlı Örtüştürme Ve Ekleme Algoritması

**YN** : Yanlış Negatif

**YP** : Yanlış Pozitif

**YSA** : Yapay Sinir Ağları

## TABLO LİSTESİ

Tablo 3.1. Karmaşıklık matrisi.....	42
Tablo 4.1. Örnek .arff dosya yapısı.....	51
Tablo 4.2. K-En yakın komşu algoritmasının başarımı.....	52
Tablo 4.3. Karar ağaçları algoritmasının başarımı.....	53
Tablo 4.4. Naive bayes algoritmasının başarımı.....	54
Tablo 4.5. Destek vektör makineleri algoritmasının başarımı.....	55
Tablo 5.1. Uygulama veri kümesi tablosu.....	58
Tablo 5.2. Temizleme işlemi yapılmamış metin dosyasının örneği.....	58
Tablo 5.3. Temizleme işlemi yapılmış metin dosyası.....	59
Tablo 5.4. Kelime kökleri bulunan metin dosyası.....	59
Tablo 5.5. Doküman-Kelime matrisi.....	60
Tablo 5.6. Benzerliklerin ve başarımın hesaplanması.....	61
Tablo 5.7. CISI türündeki dosyaların ses özellikleri.....	63
Tablo 5.8. CRAN türündeki dosyaların ses özellikleri.....	63
Tablo 5.9. MED türündeki dosyaların ses özellikleri.....	63

## ŞEKİL LİSTESİ

Şekil 1.1. Türlerle ayrılmış örnek bir sınıflandırma modeli.....	3
Şekil 2.1. Basit bir sinüs dalgası.....	8
Şekil 2.2. Ses yolunun yapısı.....	10
Şekil 2.3. Örnek bir ses tanıma sistemi.....	11
Şekil 2.4. MKS genel blok gösterimi.....	14
Şekil 2.5. MKS sistemi ayrıntılı blok gösterimi.....	15
Şekil 2.6. Eklemeli sentezleme sistemlerinde bulunan temel adımlar.....	22
Şekil 2.7. Heceleme algoritması.....	26
Şekil 2.8. MKS sisteminin genel süreci.....	27
Şekil 3.1. Gözetimsiz öğrenme.....	32
Şekil 3.2. Gözetimli öğrenme.....	33
Şekil 3.3. 10-Kat çapraz doğrulama.....	34
Şekil 3.4. Vektör uzayı modeli.....	36
Şekil 3.5. Örnek bir doküman-terim frekans matrisi.....	39
Şekil 3.6. K-Ortalama algoritması başlangıç aşamaları örneği.....	43
Şekil 3.7. K-Ortalama algoritması ideal durum örneği.....	43
Şekil 3.8. K-En yakın komşu sınıflandırması modeli.....	44
Şekil 3.9. Örnek karar ağacı modeli.....	44

Şekil 3.10. Temel YSA hücresi.....	46
Şekil 3.11. Çok katmanlı YSA modeli.....	47
Şekil 3.12. Destek vektör makinesi sınıflandırma modeli.....	48
Şekil 3.13. Lineer destek vektör makinesi sınıflandırma modeli.....	48
Şekil 3.14. Maksimal marjın ve esnek marjın.....	49
Şekil 4.1. K-En yakın komşu algoritmasının veri dağılımı.....	53
Şekil 4.2. Karar ağaçları algoritmasının veri dağılımı.....	54
Şekil 4.3. Naive bayes algoritmasının veri dağılımı.....	55
Şekil 4.4. Destek vektör makineleri algoritmasının veri dağılımı.....	56
Şekil 5.1. Metinden ses sentezleme uygulaması süreçleri.....	57
Şekil 5.2. Boyut azaltmanın başarıma etkisi.....	61
Şekil 5.3. Farklı k değerlerinin başarıma etkisi.....	62
Şekil 5.4. Doğru seslendirme süreci.....	62
Şekil 5.5. Hatalı seslendirme süreci.....	64

## ÖZET

Bu tez çalışmasında ses sentezleme konusu ile ilgili genel bir literatür taraması yapılarak, ses sentezlemenin bir çeşidi olan metinden konuşma sentezleme konusu ağırlıklı olarak ele alınmıştır.

Giriş bölümünde temel bilgilere yer verilerek literatürde ses sentezleme ile ilgili yapılan çalışmalara değinilmiştir. İkinci bölümde ses sentezleme ve çeşitleri ayrıntılı biçimde incelenmiştir. Üçüncü bölümde makine öğrenmesi konusu hakkında bilgiler verilerek bazı önemli makine öğrenme algoritmaları detaylıca açıklanmıştır. Dördüncü bölümde deneysel bir çalışma ile bazı makine öğrenmesi algoritmaları karşılaştırılmıştır. Sonraki bölümlerde metinden konuşma sentezleme için önerilen bir uygulama ile sonuçların değerlendirilmesine yer verilmiştir.

Yapılan uygulamada düz metin türlerinin monoton, robotik bir ses biçimi olarak seslendirilmesi yerine, metinleri gruplara ayırarak farklı metin türlerinin kendi alanlarına uygun olduğu düşünülen farklı ses biçimleri şeklinde seslendirilmesinin daha doğal olacağı düşünülmüştür. Bu işlem için öncelikle metinler ön işleme aşamasından geçirilerek sınıflandırma işlemi yapılmıştır. Hangi metin türünün hangi ses tonlamaları ile seslendirileceğine karar verdikten sonra, kendi kategorisine uygun olmayan metinler, sınıflandırıldığı türün ses özelliklerine göre seslendirilmiştir. Böylece varsayılan olarak belirlenen seslendirmelerin dışında bir ses tonu ile karşılaşıldığında yanlış sınıflandırma yapıldığı açık bir şekilde görülmüştür.

## **ABSTRACT**

In this thesis study, a general literature review is done about speech synthesis and text to speech synthesis subject, a type of speech synthesis, is mainly researched.

In introduction part, base information is given and studies on speech synthesis are mentioned. In second part, speech synthesis and its types are studied in detail. In third part, it's given information about machine learning subject and important machine learning algorithms are explained. In fourth part, some machine learning algorithms are compared with an experimental study. In the following chapters it is suggested to evaluate the results with an application suggested for speech synthesis.

Rather than synthesizing plain text types as a monotone, robotic sound format, it is thought to be more natural to sound different types of texts into different sound formats that are thought to be appropriate for their respective fields, by separating them into groups. For this process, the texts are first passed through the pre-processing stage and classified. After deciding which text type to synthesize with which voices, texts that are not appropriate for their category, are synthesized according to the sound characteristics of the type that is classified. Thus, when a tone other than the default voices is encountered, it is clearly seen that an incorrect classification is made.



# 1 GİRİŞ

İleri teknolojilerin kullanılabilmesi için insanlar ile bilgisayarlar arasında iletişim kurma çabaları bilim insanları tarafından uzun süredir araştırılan bir konu olmuştur (Şen K.Ö. ve ark., 2007).

İnsan-bilgisayar etkileşiminde bilgisayarların konuşması ve konuşulanları algılaması en temel niteliklerdendir. Bu konuda bilgisayar bilimleri, elektronik mühendisliği, ses mühendisliği gibi farklı alanlar bu amacı gerçekleştirebilmek için çalışmışlardır. Ses sentezleme ile ilgili iki temel çalışma alanı vardır. Bunlar; Metinden konuşma sentezleme ve konuşma tanıma sistemleri olarak isimlendirilir. Bu iki alan kendilerine özgü farklı biçimlerde incelenmektedirler.

Metinden Konuşma Sentezleme (MKS) kısaca yazılı bir metnin elektronik ortamda ses sinyallerine dönüştürülmesi demektir. Bu yazılı metin kaynağı bir metin belgesi veya elektronik kitap da olabilir, bir web sayfası da olabilir. İdeal bir MKS sisteminden insanın okuyabildiği her metni doğal insan sesi kalitesinde işleyebilmesi beklenir. Kompleks bir MKS sistemi ise sayıları kendi biçimine en uygun ses formatında okuyabilir, kısaltmaları uygun formatta seslendirebilir ve bir kelimenin farklı yazım biçimlerini ayırt edebilir (Sel, 2013). Böylece insan doğasına en uygun seslendirmenin yapılabileceği söylenebilir.

Konuşma tanıma sistemleri (KTS) ise akustik ses sinyalini bir karakter grubuna dönüştüren sistemlerdir (Aydın, 2005). Bu sistemler ile insanlar tarafından söylenenler sayısal metin formatına çevrilebilmekte ve bu metinlerde arama işlemleri gerçekleştirerek bazı örüntüler elde edilebilmektedir. Böylece konuşulanların bilgisayar tarafından algılanıp komutlar yardımıyla uygun şekilde yönlendirmeleri sağlanabilmektedir.

Makine öğrenmesi yaklaşımlarında sistemin metni anlaması, belirsiz durumların belirsizliklerini gidererek eğitilmesi esasına dayanır. Bu yaklaşımlarda, öğrenilen

durumlardan birine daha önce görülmemiş örnekler verilerek sınıflandırma yapılması istenir. Bu yaklaşımlar, eğitim materyalinin türüne, ne kadar materyale ihtiyaç duyulduğuna, kullanılan dil bilgisinin çeşidine ve üretilmek istenen çıktıya göre değişiklikler gösterir (Jurafsky D., 2008).

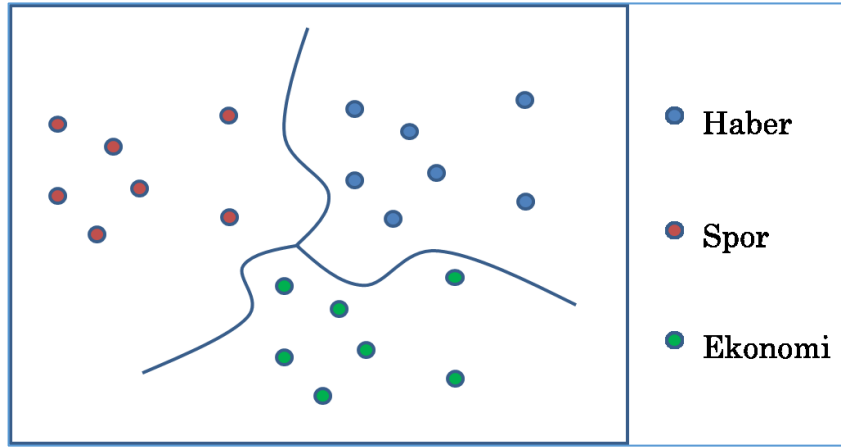
## 1.1 Tezin Amacı

Bu tez çalışması kapsamında ses sentezlemenin bir çeşidi olan metinden konuşma sentezleme konusu incelenmiş olup, bu alanda kullanılmış birçok yöntem araştırılmıştır. Metinden konuşma sentezleme için makine öğrenme metotları yardımıyla metinler çeşitli aşamalardan geçirilerek türlerine göre sınıflara ayrılmış ve her sınıfın birbirinden farklı ses biçimleri şeklinde seslendirilebilmesi hedeflenmiştir.

Metinden ses sentezleme işleminin ana hedefi, insan sesine en uygun ses davranışının elde edilmesidir. Düz bir metinden en uygun ses biçiminin elde edilmesi, metnin doğru metotlarla işlenmesi ve yorumlanmasına bağlıdır. Bir düz metnin nasıl tonlanması gerektiği, yazım-ses farklılıklarının ayrt edilmesi, özellikle dilbilim/anlambilim ve sinyal işleme alanları kapsamındadır. Bilgisayar mühendisliği alanında ise veri madenciliği, örüntü tanıma, doğal dil işleme, makine öğrenmesi ve otomatlar gibi alt alanlardan faydalanılması metni anlamlı şekilde yorumlamamıza yardımcı olmaktadır. Ancak bu alanda birçok zorluk karşımıza çıkmaktadır. Bunlardan en önemlisi vurgu/tonlama işleminin zorluğudur.

Bu çalışmada, ham metinler önce metin ön işleme sürecine tabi tutularak sayısal değerlere dönüştürülmüş ve bu sayısal (vektörel) değerlerin makine öğrenmesi algoritmaları yardımıyla başarıyı en yüksek seviyede sınıflandırılması gerçekleştirilmiştir. Gözetimli öğrenme modelinde problem, sınıflandırma problemi olarak ele alınmakta ve eğitilen sistem test veri setine göre tahmin ve tanıma amacıyla kullanılmaktadır. Sınıflandırma sonucunda her sınıf için başlangıç olarak belirlenen ayrı ses dosyaları elde edilerek insan doğasına uygun seslendirme yapılabilmesi istenmiştir. Farklı kategorilere ait metin türlerinin farklı ses biçimleri şeklinde seslendirilmesinin insan doğasına daha uygun olacağı düşünülmüştür. Kısaca, metinden konuşma sentezleme işlemi bir metin sınıflandırma problemi olarak ele alınmıştır. Örneğin; haber içerikli metinler belirli bir formatta seslendirilirken, spor içerikli metinler daha farklı bir formatta seslendirilmek

istenmiştir. Alan türlerine göre ayrılmış örnek bir sınıflandırma modeli Şekil 1.1’de gösterilmiştir.



Şekil 1.1. Türlerine ayrılmış örnek bir sınıflandırma modeli

Tüm bu işlemler yapılırken veri madenciliği ve makine öğrenmesi algoritmaları kullanılarak en uygun sınıflandırma yöntemi uygulandıktan sonra ses dosyaları elde edilmeye çalışılmıştır.

## 1.2 Tezin Kapsamı

Bu tez kapsamında ses sentezleme alanının alt dallarından konuşma tanıma sistemlerinden ziyade metinden konuşma sentezleme konusu üzerinde durulmuştur. Bu doğrultuda özellikle metinler makine öğrenmesi teknikleri kullanılarak türlerine ayrılmış, bu türlere uygun seslendirme yapılabilmesi için en uygun metotların tanımlanmasına çalışılmıştır. Ele alınan metin türleri akademik dil ile yazılmış olan makale örneklerinden oluşmaktadır. Belli dilbilgisi kurallarına uygun olmadan yazılan metinler bu tez kapsamına dâhil edilmemiştir.

## 1.3 Ses Sentezleme Üzerine Yapılan Araştırmalar

### 1.3.1 Mekanik Ses Sentezleme Sistemleri

Ses sentezi sistemleri başlangıçta mekanik çalışmalarla başlamıştır. Yapılan araştırmalar sonucunda 1779 yılından bugüne dek konuşma sentezleme konusunda birçok çalışma olduğu görülmektedir (Eker, 2002). Bu alandaki ilk araştırma 1779 yılında Rus Profesör Christian Kratzenstein tarafından yapılmıştır. Christian

Kratzenstein beş uzun sesli harfin (/a/, /e/, /i/, /o/, ve /u/) yapılarını inceleyerek kısmen bu sesleri üretebilen bir model icat etmiştir (Sel, 2013).

1791 yılında, Wolfgang von Kempelen bir makine geliştirmiş ve bazı sesleri bu makine ile elde etmeyi başarmıştır. “Konuşma Makinesi” adını verdiği bu makine, insan gırtlak yapısının mekanik cihazlar ile benzetilmesi ve konuşma esnasında çıkarılan seslerin bu cihazlar ile taklit edilmesi esasına dayanmaktadır (Yılmaz A. E., 2009).

1800’lü yıllarda Charles Wheatstone, Kempelen’in cihazını geliştirerek daha iyi sonuçlar elde etmiştir. Geliştirdiği bu makine birçok sesi üretebilmekte, hatta bazı kelimeleri tam olarak sentezleyebilmektedir. 1838 yılında Willis bazı sesli harfler ile ses yolunun yapısı arasında bağlantı kurmuş, ses yolunun yapısına benzeyen borular ile bazı sesli harfleri seslendirebilmiştir. Willis seslendirilen sesin kalitesinin borunun çapından ziyade, sadece uzunluğuna bağlı olduğunu bulmuştur (Güldalı, 2009).

1950’li yıllara kadar, operatör ile yönlendirilerek veya otomatik olarak önceden kaydedilmiş sesleri sırayla sentezleyen çeşitli mekanik cihazlar geliştirilmiştir (Klatt, 1987). Mekanik ve yarı elektriksel sistemler üzerine yapılan çalışmalar 1960’lı yıllara kadar sürmüştür fakat bu alanda önemli bir başarı elde edilememiştir. (Lemmetty, 1999).

### 1.3.2 Elektrikli Ses Sentezleme Sistemleri

Bilinen ilk elektriksel ses sentezleyici 1922 yılında Stewart tarafından tanıtılmıştır. Bu sentezleyici ile bir elektrik zili ve ses yolu için iki rezonans devresi kullanarak ses üretilmiştir. 1930’lu yıllarda Bell Laboratuvarlarında çalışan bilim adamları konuşma sesinin temel ton ve rezonanslarını otomatik olarak analiz eden “Vocoder” cihazını geliştirmişlerdir (Sel, 2013). Elektrikli tarzdaki bir başka ses sentezleyici Homer Dudley tarafından 1939 yılında sunulan VODER (Voice Operating Demonstrator) cihazıdır. Bu çalışma, oluşturulan ses kalitesi düşük düzeyde olsa da yapay ses oluşturma açısından önemli bir adım olarak nitelendirilmiştir. (Flanagan J. L. ve ark., 2008). 1940’ların sonuna doğru Dr. Franklin S. Cooper ve meslektaşları tarafından imal edilen makine ile sesin akustik desenlerine ait resimleri bir spektrogram şeklinde yeniden sese dönüştürülmüştür. 1950’li yıllara kadar, operatör ile idare edilerek veya

otomatik olarak önceden kaydedilmiş sesleri art arda sıralayarak sentezleyen çeşitli mekanik cihazlar geliştirilmiştir. Bilim dünyasına ilk sunulan format sentezleyici konuşma oluşturma sistemi olan PAT (Parametric Artificial Talker) Walter Lawrence tarafından 1953 yılında geliştirilmiştir (Ünaldı, 2007). İlk bilgisayar temelli ses sentezleme sistemi ise 1950'lerin sonunda üretilmiştir. 1968 yılında Japonya'da Noriko Umeda ve arkadaşları tarafından ilk İngilizce metinden ses sentezleme sistemi geliştirilmiştir. Bu sistemde sentezlenen ses, şimdiki sistemlerin kalitesinde olmasa da anlaşılabilir biçimde üretilmiştir (Sel, 2013).

1960 sonrasında ise bilgisayar teknolojisinin kullanımı ile metinden konuşma sentezleme çalışmaları büyük bir ivme kazanmıştır. İlk yıllarda, başta İngilizce olmak üzere genelde Hint-Avrupa dil ailesindeki diller üzerinde yoğunlaşan çalışmalar, zamanla diğer dil ailelerine de uygulanmıştır (Yılmaz A. E., 2009).

### 1.3.3 Türkçe Ses Sentezleme Sistemleri

İlk başta İngilizce olmak üzere Türkçe de dâhil birçok dil için hazırlanan MKS sistemleri, ticari ürünler olarak müşterilere sunulmuştur (Eker, 2002). Sunulan ticari sistemler dışında Türkçe Metinden Konuşma Sentezleme (TMKS) alanında birçok akademik çalışmanın da yapıldığı görülmektedir (Yılmaz A. E., 2009).

MKS sistemleri genel olarak detaylandırıldığında 3 farklı MKS yöntemi ile karşılaşılır (Uslu, 2010). Bu yöntemler şu şekildedir:

- Kural Tabanlı Sentezleyiciler
- Söyleyiş Sentezleyiciler
- Eklemeli Sentezleyiciler

TMKS alanında yapılan akademik çalışmalar incelendiğinde oluşturulan birçok sistemde eklemeli sentezleme yöntemi kullanılmıştır (Eker, 2002). Türkçe dili eklemeli bir dil olduğu için kelimeler hecelerle birleşmesiyle oluşmaktadır. Ayrıca bir kelimenin ek olarak farklı kelimelere türetilmediği görülebilmektedir (Şentürk T. ve Adalı E., 2010). Bu yüzden TMKS sistemleri için en uygun yöntemin 'Eklemeli Sentezleme' olduğu belirtilmektedir (Aşlıyan R. ve ark., 2006).

Eklemeli sentezlemenin kaliteli olması seslerin ön işleme ve birleştirme algoritmasına bağlıdır. Birleştirilecek seslerin frekanslarının uyum içerisinde olması

sistemin başarısına doğrudan etki etmektedir (Sel, 2013). Yapılan arařtırmalar sonucunda seslerin birleřtirilmesi için Örtüşme-Ekleme Algoritması, Eşzamanlı Örtüşme-Ekleme Algoritması, Zaman-Alan Senkronize Ton Örtüşme ve Ekleme Algoritması ve Dalga Şekilli-Benzerlik Tabanlı Senkronize Örtüşürme ve Ekleme Algoritması gibi farklı algoritmalar olduđu görölmektedir (Verhelst, W. and Roelands, M., 1993).

Harflerin okunma süreleri, kullanılma sayıları ve metnin karmaşıklığı gibi ölçütler de sentezlenen konuşmanın doğallığını etkilemektedir (Canal Ş.M. ve ark., 2010). Canal ve arkadaşları, Türkçe metinden konuşma sentezleme konusunda doğallığın artırılmasına yönelik yaptıkları çalışmalarda çeşitli çözüm yöntemleri ile insan sesine yakın bir metinden konuşma sentezleme sistemi geliřtirmişlerdir. “Ünlüden-Ünlüye” ses sentezleme yöntemini kullanan bu sistem, girdi olarak yazılı metni almakta, standartlařtırmakta ve çıktı olarak bu metne karşılık gelen Türkçe sesleri üretmektedir. Bu sistem ile seslerin birleřtirilmesinin daha kolay olduđu ve dijital olarak çok fazla işleme ihtiyaç duyulmadığını gözlemlemişlerdir. Ancak bu yöntemde kullanılacak ses gruplarının miktarı, ikili ses birleřtirme yöntemlerinde kullanılan ses gruplarının miktarından daha fazla olduđu için daha büyük bir veri tabanına ihtiyaç duyulmaktadır.

Şayli (Şayli Ö., 2002), Türkçe MKS sistemleri için süre tabanlı bir model üzerinde çalışmış, fonem ve trifon tabanlı incelemelerin sonucunda ortalama süreleri baz almıştır.

Öztürk (Öztürk Ö., 2005), fonemler için süre tabanlı ve temel frekans eğrilerini esas alıp istatistiksel olarak fonemin türü, hecelerinin sayısı, konumu ve vurgulu olup olmaması gibi özellikleri inceleyerek analiz yapmıştır. Bunun sonucunda ortalama sürenin en etkili parametre olduğunu rapor etmiştir.

Oskay (Oskay B. ve ark., 2001), cümle bazlı temel frekans eğrilerinin genelleřtirilmesine çalışarak, F0 eğrileri, olumlu, olumsuz ve soru cümleleri için doğrusal ve ikinci derece fonksiyonlar ile modelleme yapmışlardır.

Külekçi ve Oflazer (Külekçi M. O. ve Oflazer K., 2006), metin içerisindeki söz gruplarını belirlemeye çalışmışlar, bunlara 3 kademeli ezgi seviyesi atamışlardır (Uslu İ.B. ve ark., 2011).

Türkçede fiil çekimleri, bazı istisnalar haricinde, kurallara bağlıdır (Aydemir T. ve Yılmaz, 2010). Aydemir ve Yılmaz çalışmalarında fiillerin otomatik çekimlenmesi ve vurgularının belirlenmesi üzerinde çalışmışlardır.

Uslu (Uslu İ.B. ve ark., 2011), fiil çekimlerini süre, perde frekansı ve enerji gibi akustik özellikler ile tanımlayan bir MKS ara yüzü önerisi getirmişlerdir.

Yılmaz (Yılmaz A. E., 2009) tarafından Türkçe metinden konuşma sentezleme uygulamaları için altyapı sağlayacak olan bir veri sözlük seti tanıtılmıştır. Çalışmada, geliştirilen tümleşik ürünün hem kuramsal, hem de uygulamaya dönük ihtiyaçlara cevap verecek geniş yelpazede bir işlevsellik sağlayacağı düşünülmüştür.

Aşlıyan ve Günel (Aşlıyan R. ve Günel K., 2008) Türkçe metinlerin insan sesine dönüştürülmesi suretiyle bir konuşma sentezleme sistemi geliştirmiştir. Bunun için en küçük ses birimi olarak Türkçe dilinin doğal yapısı gereği heceleri kullanmışlar, TASA algoritması yardımıyla Türkçedeki farklı heceleri tespit edip kaydetmişlerdir. Bundan yola çıkarak önışlem süreci sonrası hece-ses veri tabanı oluşturmuşlar, bu ses veri tabanı kullanılarak vurgu ve tonlama özellikleri bakımından zayıf olsa da art arda bağlama yöntemi ile Türkçe metin sentezleme işlemi gerçekleştirmişlerdir.

## 1.4 Varsayımlar

Tez genelinde metinden konuşma sentezleme kuralları genel çerçevede ele alınmasına rağmen uygulama gerçekleştirme bölümünde metin türleri için İngilizce makale örnekleri kullanılmıştır. Belli bir akademik dil düzeninde yazılmamış olan metin türleri tez kapsamına dâhil edilmemiştir. Makaleler içerisinde yer alan metinlerdeki her kelime için ayrı seslendirme yapılmasından ziyade, her makale türü için metnin geneline bakılmak suretiyle sabit bir seslendirme yapılması hedeflenmiştir. Kullanılan üç farklı tür metin belgesi için başlangıç ses değerleri belirlenerek uygun görülen ses tonlama ve biçimleri MBROLA ses veri tabanından seçilerek kullanılmıştır. Makine öğrenmesi teknikleri uygulandıktan sonra veri kümesinde başlangıçtaki varsayılan ses biçimlerinin dışında oluşan ses dosyaları hata olarak kabul edilmiştir. Elde edilen bu ses dosyaları aynı zamanda uygulanan makine öğrenmesi algoritmasının başarımına etki etmektedir.

## 2 SES SENTEZLEME

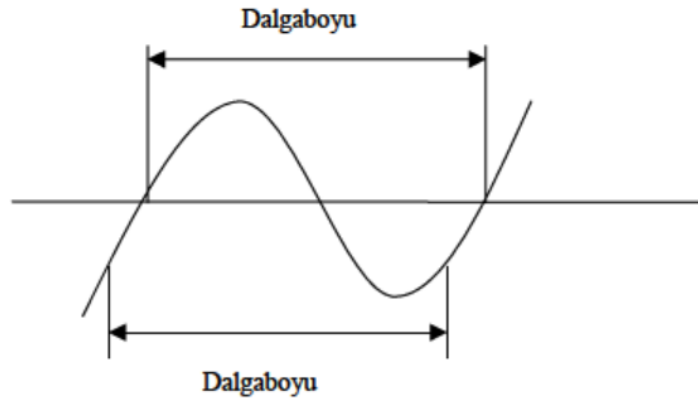
### 2.1 Genel Bilgiler

#### 2.1.1 Sesin Yapısı

Ses, insan kulağında işitme duyusu oluşturan hava moleküllerinin titreşim meydana getirerek hava basınç değişimleri şeklinde ortaya çıkan işitsel izlenim olarak tanımlanır (Aydın, 2005).

Sesin yapısı ile ilgili kavramlardan bazıları şunlardır:

Ses Dalgası: Ses sıvı, katı, gaz ortamlarda yaklaşık 20 Hz ile 20 KHz arasındaki insan kulağının algılayabileceği basınç değişiklikleri olarak tanımlanmaktadır. Bu frekans aralığındaki mekanik dalgalar işitme duyumuzu uyarırlar ve bizim için özellikle önemli olan ses dalgalarını oluştururlar. İnsan kulağına bir ses dalgası geldiğinde kulak ses dalgasındaki basınç değişikliklerini sinirler kanalıyla alıp beyne iletir. En basit ses dalgası bir frekansa ve sabit bir genliğe sahiptir. Buna sinüs dalgası adı verilir. Aşağıda basit bir sinüs dalgası grafiği görülmektedir (Aydın, 2005).



Şekil 2.1. Basit bir sinüs dalgası



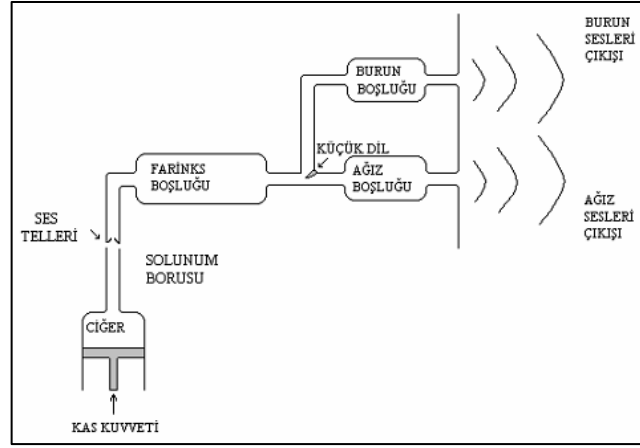
Frekans: Sinüs dalgasındaki iki tepe arasında kalan uzaklık dalga boyudur. Bir saniyede gözlenen dalga tepesi sayısına frekans denir. Frekans ile sesteki tizlik belirlenir. Birim olarak saniyedeki çevrim (Cycle Per Second-CPS) veya Hertz (Hz) ile ifade edilir. Düşük frekanslara bas sesler, yüksek frekanslara ise yüksek seviyeli sesler denir.

Genlik: Sesin bir başka karakteristik özelliği genliktir. Sesler yumuşak sesler veya yüksek sesler şeklinde basınca bağlı olarak değişir. Bu değişim ile havada havayı sıkıştırmak için kullanılan güce bağlı olan bir tür küçük veya büyük basınç değeri elde edilir. Ses gücü veya seviyesi için desibel (dB) birimi kullanılmaktadır. Kulağın algılama özelliği ile ilgili yapılan çalışmalarda ses gücünün artması ile hissedilen ses artışının doğrusal olmadığı ve logaritmik bir ses şiddeti ile duyma olduğu anlaşılmıştır. Bu sebeple algılanan ses logaritmik olduğundan haberleşme sistem ve cihazlarının yapısı ve ölçü birimleri logaritmik olarak düzenlenmiştir (Aydın, 2005).

Gürültü: Periyodik olmayan titreşimlere gürültü denir. Bunlar kulağın, teknik duyumu bakımından sınırları zorlayan ve psikolojik rahatsızlıklar doğuran bozuk ses biçimleridir (Aydın, 2005).

### 2.1.2 Konuşma ve Özellikleri

Ses dalgası, ses üretim sistemini meydana getiren anatomik yapıların istemli hareketleri ile oluşan akustik bir basınç dalgasıdır. Bu sistemde ciğerler, nefes borusu, gırtlak, boğaz, ağız boşluğu ve burun boşluğu yer alır. Teknik terim olarak boğaz ve ağız boşluğu 'ses yolu' olarak ifade edilir. Böylece ses yolu, gırtlak çıkışından başlayıp dudaklarda sona erer denebilir. Burun yolu ise damaktan başlayıp burun deliklerinde son bulur. Sesin üretimi için kritik olan anatomik yapıları, ses telleri, damak, dil, dişler ve dudaklar şeklinde sıralayabiliriz. Ağız iyice açıldığında ağız boşluğunun arka tarafında duran damağın yumuşak uzantısı görülmektedir. Buna 'küçük dil' adı verilir. Ses yolunu oluşturan bu anatomik yapılar, farklı pozisyonlar alarak değişik sesleri oluştururlar. Ses yolunun yapısı aşağıdaki şekilde görülmektedir (Aydın, 2005).



Şekil 2.2. Ses yolunun yapısı

## 2.2 Ses Tanıma

Ses tanıma, akustik sinyalin bir karakter grubuna dönüştürüldüğü sistemdir. Son zamanlarda oldukça popüler bir sistem olup, günümüzde geniş bir kullanım alanına sahiptir. Tanıma işlemi bazı farklı teknolojilerin birlikte kullanımı ve uygulanması ile gerçekleştirilir.

Gerçekleştirilmek istenen ses tanıma sistemleri sinyal işleme, akustik, model tanıma, haberleşme ve bilgi teorisi, dilbilim, psikoloji ve bilgisayar bilimi gibi birçok farklı disiplini içerir. Ses tanımda kullanılan bu disiplinler aşağıdaki amaçları yerine getirmektedirler (Rabiner L. R., Juang B. H., 1993).

**Sinyal İşleme:** Bir ses sinyalinden gerekli bilgiyi en etkili ve en uygun şekilde çıkarma işlemi gerçekleştirir.

**Fizik:** İnsan sesinin akustik ve fiziksel yapısı ile ilgilenir (konuşma sistemi ve duyma mekanizması gibi).

**Model Tanıma:** Veriyi prototip olan modellerde gruplamak ve bir çift modelin özelliklerini temel alarak eşlemek için kullanılır.

**Haberleşme ve Bilgi Teorisi:** Bazı ses modellerini belirleyen metotları içerir.

**Dilbilim:** Dil içinde kodlanan ifadelerin sessel, morfolojik, sentaktik, semantik ve pragmatik düzeylerdeki yapısını, işlevlerini ve bu düzeyler arasındaki ilişkileri inceler.

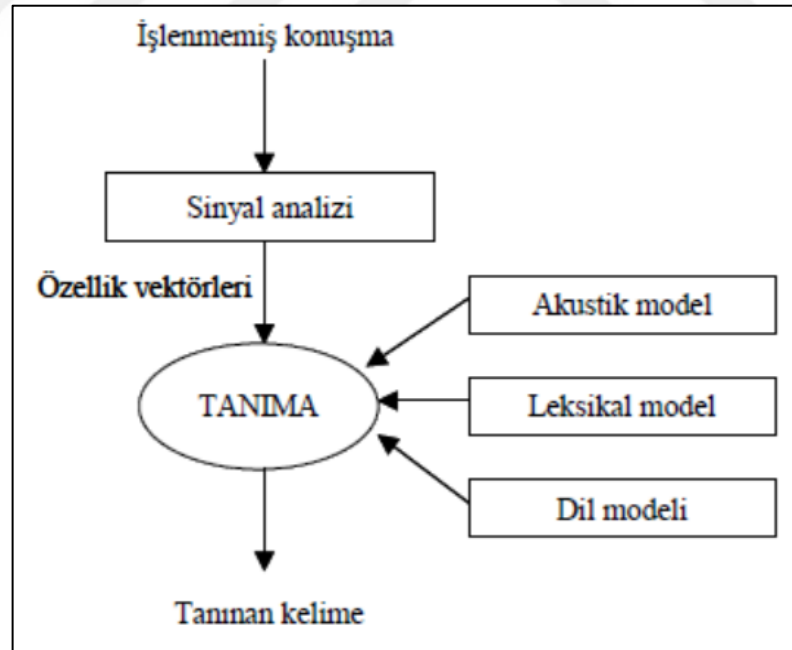
Fizyoloji: Ses tanıma çalışmalarında insan sinir sisteminde sesi anlama ve sesi üretmedeki mekanizmayı açıklamak için kullanılır.

Bilgisayar Bilimi: Yazılım ve donanımı ses tanımada en verimli yöntemin uygulanması için etkili algoritmaları oluşturmak ve uygulamak amacıyla kullanılır.

Psikoloji: Basit görevlerde, insanlar tarafından kullanılan teknolojiyi etkin hale getiren faktörleri anlamakta kullanılır (Aydın, 2005).

### 2.2.1 Ses Tanıma Süreci

Ses tanıma işlemi, kullanılan dilin sözlükteki bir kelimesinin söyleniş biçimine karşılık gelen özellik vektörleri dizisinin haritalanmasıdır. Konuşma esnasındaki özellik parametreleri belirlendikten sonra bu parametreler için istatistiksel bir model bulunur. Buna akustik modelleme denir (Yaşaroğlu, 2003). Konuşmacıların söyledikleri konuşma sinyalleri akustik model veri tabanına kaydedilir. Bütün bu veri tabanında arama ve verilen bir sinyal için sözlükteki en iyi eşleşmeyi seçme vasıtasıyla ses tanıma işlemi gerçekleştirilir.



Şekil 2.3. Örnek bir ses tanıma sistemi

Ses tanıma sürecinde konuşmacı tarafından söylenenler sisteme ses biçimleri şeklinde girilerek kayıt altına alınır. Böylece, henüz işlenmemiş konuşma verisi elde edilir. Sonraki aşamada ses sinyali sayısal veri biçimine dönüştürülür.

Sayısallaştırılan ses sinyali üzerinde sinyal işleme teknikleri uygulanarak sesin ayırt edici özellikleri ortaya çıkarılır. Daha sonra sinyal işleme sonucunda oluşan özellik vektörleri bazı teknikler ile modellenir. Modellemenin sonucunda kelimenin söylenişine karşılık gelen özellik vektörleri dizisinden bir kelime modeline ulaşılır ve bu model veri tabanında bulunan hazır şablonlar ile karşılaştırılır. Ses tanıma işlemi, bu veri tabanında arama ve verilen sinyal için en iyi eşleşen kelimeyi seçme ile son bulur (Aydın, 2005).

### 2.2.2 Ses Sentezleme ile Neler Yapılabilir?

Ses sentezleme ile, örneğin Türkçe olarak söylenen herhangi bir ses, sayısal metin haline çevrilebilmekte ve yazılan herhangi bir Türkçe metin seslendirilebilmektedir. Genelde santrallerde ve telefonla müşteri hizmetlerinde kullanılan bu sistemin başka pek çok ticari uygulaması vardır. Bir müşteri hizmetleri servisine telefon edildiğinde, robotik olarak müşteri ile konuşan ve konuşarak komut alan robotlar genelde bu sistemi kullanmaktadırlar.

Ses tanıma sistemi ile aynı zamanda telefon dinleyerek konuşmalar analiz edilebilir ve bu analiz sonucuna göre değişik sistemlerde kullanılabilirler. Buna örnek olarak polis tarafından telefon dinleme ve dinlenen sesleri detaylı analiz etmek üzere otomatik olarak bir veri tabanına kaydetme işlemi gösterilebilir. Konuşmalar metne dönüştüğünde metin tarayıcı programlar ile çok kolay biçimde metin içinde aramalar yapılarak konuşma hakkında çeşitli örüntüler de çıkarılabilmektedir.

### 2.2.3 Ses Tanımının Kullanım Alanları ve Kısıtları

Ses tanıma yazılımları konuşulan kelimeleri yazılı metinler haline çevirmekte ve böylece bilgisayar kullanıcılarına yardımcı olmaktadır. Bu yazılımlar birçok kullanıcı tarafından farklı ihtiyaç ve amaçlarla kullanılmaktadır. 3 tip kullanıcıdan bahsedilir (Fransson vd., 2001). Bunlar;

1) Ellerini kullanma engeli olan kullanıcılar: Doğru bir biçimde konuşabilen fakat yazı yazmak için ellerini kullanmakta zorlanan insanlardır. Bunlar bazı sinirsel problemleri olan, bir kaza geçirmiş, zayıf durumda olanlar veya sağlıklı yazamayan kullanıcılarıdır.

2) Profesyoneller: Yazı yazmak için zamanı olmayanlar veya iyi yazamayanlardır. Bu gruptaki bazı çalışan insanlar tıbbi ve kanuni alanlarda çalışanları veya yazılmış yayınları takip etme ihtiyacı olanları içerir. Yazılım kullanıcılara maddi yarar sağlayabildiği gibi kolayca rapor almak için zaman da kazandırır. Tipik kullanıcıları doktorlar, avukatlar, psikologlar, satış sorumluları ve diğerlerinden oluşur.

3) Öğrenme zorluğu olan kullanıcılar: Doğru olarak yazmalarını önleyen öğrenme yetersizliği olan insanlardır.

Ses tanıma uygulama safhasında bazı problemler vardır. Bunlardan bazıları, her insanın konuşma tarzının ve ağzının farklı olması, yani bir standardın olmaması olarak sayılabilir. Lehçe tanınsa bile ses tanıma aracı herkes için düzgün çalışmayı garanti etmez. Bazı ses tanıma uygulamaları sessiz ortamlarda bile tanıma yapamayıp doğru sonuç vermez iken, diğerleri en gürültülü ortamlarda dahi verimli kullanılabilirler.

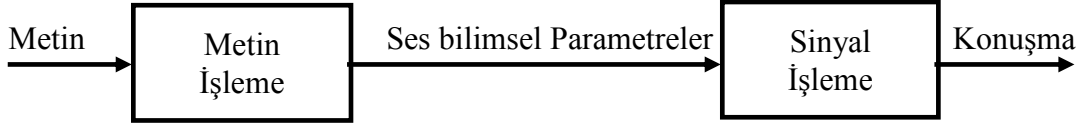
Kısıtlarını maddeler halinde özetleyecek olursak:

- Sessiz, kontrollü ortamda iyi çalışırlar. Fakat gürültülü ortamlarda başarımları yüzdeleri düşük olabilir.
- Basit ses tanıyıcılar kelimeleri duygusuz biçimde söylerler.
- Büyük sözlük kullanıldığında karmaşıklık artar.
- Yüksek miktarda işlem gücü ister.
- Kısa kelimeleri ayırt etmenin ayrı zorlukları vardır.
- Herkes için çalışma garantisi yoktur.
- Bazı tanıyıcıların diğerlerine göre işleme tarzı daha zahmetlidir.
- En iyi durumda kelime doğruluğu %95'lere ulaşır.

## 2.3 Metinden Konuşma Sentezleme

### 2.3.1 Metinden Konuşma Sentezleme Nedir?

Metinden Konuşma Sentezleme (MKS), bilgisayar tarafından yazılı bir metnin ses sinyallerine dönüştürülme işlemidir. MKS uygulamaları çoğunlukla çoklu ortam araçlarında kullanıcı ile olan etkileşimin üst düzeylere artırılması amacıyla kullanılmaktadır (Sel, 2013). Sistem blok olarak aşağıdaki gibi gösterilebilir.



Metinden konuşma sentezlemenin ana görevi, bir dizi metin kelimesini girdi olarak almak ve çıktı olarak akustik bir dalga formu üretmektir. Bu sistemin seslendireceği her kelime için bir telaffuza ihtiyaç vardır (Jurafsky D., 2008). Farklı ortamlarda farklı şekilde üretilen bu ses sisteminin dilbilgisi ile nasıl ilişkili olduğunu açıklayan alan sesbilim alanıdır.

Özetle MKS sistemi yazı biçimindeki veriyi (text) girdi olarak alır ve konuşma diline uygun çıktı üretir.

### 2.3.2 MKS Kullanım Alanları ve Yararları

Metinden konuşma sentezlemenin kullanım alanlarına örnek olarak engelli insanlara yardım sistemleri, çoklu ortam cihazları, navigasyon uygulamaları, tüketici elektronik ürünleri, telekomünikasyon sistemleri vb. verilebilir (Canal Ş.M. ve ark., 2010). MKS sistemleri görme engellilere bilgisayar kullanım kolaylığı sağladığı gibi internet üzerindeki metin bölümlerine erişip normal bir kullanıcının kullanabildiği birçok uygulamayı kullanabilme olanağı sunar. Ayrıca her metni okuyarak ve yazılanları kontrol ederek kullanıcıları doğru yönlendirebilir. Birçok alanda kullanılan ve geliştirilen bu sistemler sabit veya taşınabilir cihazların üzerinde kullanılabilir (Sel, 2013).

MKS'nin diğer bir kullanımı telefon bağlantısı ile bilgisayar sistemlerine uzaktan erişim sağlamasıdır. Bu tür kullanımlar rezervasyon sistemlerinde sıkça kullanılan bir yöntemdir. Bankacılık ve finans şirketleri bu teknoloji sayesinde müşterilerine

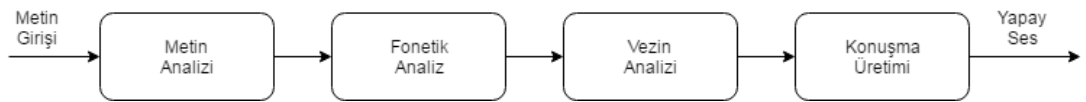
hesap bilgilerini sağlayarak telefon hattı vasıtasıyla yeni yönlendirme işlemleri yaptırabilirler. Bu kolaylıklar müşterileri basit işlemler için bankaya giderek sıra bekleme zahmetinden kurtarır. Tüm bu olanaklardan yararlanmak için özel bir donanım gerekmemektedir. Dünyanın herhangi bir yerinde bulunan bir telefon ile bankacılık işlemlerine kolaylıkla ulaşılabilir (Eker, 2002).

MKS sistemleri eğitim alanında da kullanılmaktadır. Eğitimciler bu teknolojiye konuşma sentezleyicisi bulunan bir bilgisayar yardımı ile birçok farklı dilin yazılışını ve telaffuzunu öğretmek için yararlanabilir (Sel, 2013).

### 2.3.3 MKS Süreci ve Yöntemleri

MKS sistemleri genelde iki ana bölümden meydana gelmektedir. Bunlar metin işleme ve sinyal işleme olarak isimlendirilir. Metin işleme bölümü ile sentezlenecek dilin yapısına uygun bazı ön işlemler vasıtasıyla metin hakkında detaylı bilgiler elde edilir. Bu bilgiler kullanılarak ses biçimleri elde edilmeye çalışılır. Örneğin varsa metindeki kısaltmalar, sayılar, tarihsel gösterimler ('Doç.', '2.', 'T.C.', 'Mah.', vb.) doğru okunuş biçimlerine çevrilmelidir. Sinyal işleme bölümünde ise ses bilimsel parametreler kullanarak konuşma elde edilmeye çalışılır (Uslu, 2010).

Yazılı bir metinden konuşma sentezlemenin başlıca adımları aşağıdaki gibidir;



Şekil 2.5. MKS sistemi ayrıntılı blok gösterimi (Mahwash, A. ve Shibli, N., 2014)

Bu adımlar aşağıda detaylıca açıklandığı gibi gerekli işlemlerden geçirilerek ses dosyalarının elde edilmesi sağlanır.

- Metin / Yapı Analizi (Text / Structure Analysis)

Metin analizi bölümünde yazılı metin yapısal olarak işlenir ve paragrafların nerede başlayıp bittiğine, cümlenin nerede başlayıp nerede sonlandığına karar verilir. Daha sonra metin ön işleme (text pre-processing) aşamasında noktalama işaretleri ve tarih biçimleri belirlenir. Bu işlemler dilin özel yapısına uygun olarak gerçekleştirilir. Örneğin İngilizcede kısaltmalar, tarihler, zamanlar, numaralar, para birimleri, e-posta

adresleri ve diğer özel gösterim türleri için özel bir işleyiş gereklidir. Diğer diller de buna benzer veri türlerini kendilerine özgü bir biçimde ele alır ve işler.

- Fonetik Analiz (Text-to-phoneme conversion)

Fonetik analizde her bir kelime ses birimlerine çevrilir. Ses birimi (hece-phoneme) bir dildeki seslerin en küçük birimidir. İngilizcede yaklaşık 45 adet ünlü-ünsüz ses birimi bulunur. Örneğin, “times” sözcüğü 4 adet ses biriminden yararlanılarak (t ay m s) seslendirilir. Farklı diller farklı ses birimlerine sahiptir.

- Vezin (Ölçü) Analizi (Prosody Analysis)

Vezin analizi, cümle yapısı süreci ile kelime ve sesleri cümle için uygun olan vezni (aruz) bulmak için kullanılır. Vezin, kelimenin söylenirken ağızdan çıkan seslerden daha fazla konuşma özelliği içerir. Bu özellikler, ses perdesi (pitch or melody), zamanlama (timing or rhythm), duraksama (pausing), konuşma oranı (speaking rate), kelimeler üzerindeki vurgu ve daha başka birçok özellik içermektedir. Doğru vezin, doğru konuşma seslerini belirleyerek doğru anlamı verebilmek için önemlidir.

- Konuşma Üretimi (Waveform Production)

Konuşma üretimi bölümünde fonem ve vezin bilgisi her bir cümle için ses dalgası üretmek amacıyla kullanılır. Fonem ve vezin bilgisinden ses dalgası üretmenin daha birçok yolu vardır. Genel olarak bu işlem iki yolla yapılır. Bunlardan biri; daha önceden hazırlanmış kayıtlı insan sesi parçalarını birleştirmek, diğeri sinyal işleme tekniklerini kullanmak şeklindedir.

#### 2.3.4 MKS’de Karşılaşılan Problemler

MKS sisteminde karşılaşılan problemlerden biri, söyleyişte farklılık oluşturmaktır. Bu durum her dil için kendine özgüdür ve çözümü kullanılan dile uygun olmalıdır. Metinden en uygun ses biçiminin elde edilebilmesi için üç aşamalı olarak dönüştürme işlemi yapılır (Güldalı, 2009). Bunlar;

- Metin ön işleme
- Dilbilimsel çözümlemede veri çıkarımının söyleyiş için doğru yapılması
- Ölçü analizinin doğru tonlama, vurgu ve süre açısından uygun olması



şeklinde özetlenmektedir.

Metin önışleme aşamasında cümle veya paragraf içerisinde yer alan noktalama biçimlerinin kısaltma olarak kullanıldığı durumlarda, cümle veya paragraf sonu gelmiş gibi algılanabilir. Bu durumda cümle veya paragrafın nerede başlayıp nerede bittiği doğru tespit edilemez. Örnek olarak; cümle içerisinde “T.C.” kısaltması ile karşılaşıldığında, kullanılan nokta işareti yanlış çevrim yapılarak bir cümle sonu olarak algılanabilir. Bu durumların üstesinden gelmek için cümle ve paragraflar açık bir biçimde işaretlenerek böyle yapısal hatalardan uzak durulabilir.

Sentezleyicinin bütün cümle ve kelime kısaltmalarını, tarih ve zaman biçimlerini bilme ihtimali yoktur. Örneğin: cümlede geçen 6/2 ifadesi 2. ayın 6’sı şekline dönüştürülmesi yerine sentezleyici bunu 6/2 matematiksel işleminin sonucu olarak ele alabilir. Ayrıca İngilizcedeki yıl olarak ifade edilmek istenen 1987 senesi “nineteen eighty seven” olarak seslendirilmesi yerine sayısal bir ifade olarak “one thousand nine hundred eighty seven” olarak seslendirebilir. Bu durumlar anlam karmaşası oluşturabilir.

Sentezleyiciler alışlagelmiş çoğu kelimeyi doğru olarak seslendirebilir ancak farklı ve ilk defa karşılaşılan sıra dışı isim, kurum adı vb. özel durumlarda yazılışları aynı, okunuşları farklı sözcüklerin tahmin edilmesinde oldukça zorlanırlar. Bunlar fonetik telaffuz kullanılarak önlenebilir.

Fonetik olmayan bir dil yazıldığı gibi okunmaz ve MKS sisteminin bu telaffuzları veri tabanında tutması gerekebilir. Fakat bu durum kayıt işlemi için ayrı bir zaman harcanmasına ve maliyet artışına sebep olur. Bu duruma farklı bir çözüm olarak telaffuz için genel kurallar üretmek şeklinde bir çözüm sunulabilir. Daha sonra bu kurallar girdi olarak kelimelere uygulanır. Bu yöntem programlama mantığına uygun bir çözüm olsa da iyi derecede dilbilimsel çalışma gerektirir. Çünkü her kural tüm kelimelere uygulanamayabilir ve bu istisnai olan durumlarda başarısızlıkla sonuçlanabilmektedir (Eker, 2002).

Telaffuz açısından bir diğer problem ise telaffuz farklılıklarıdır. Bir kelime için farklı telaffuzlar varsa metin işleme aşamasında hangi telaffuzun daha doğru olduğuna karar verilemeyebilir (Yılmaz A. E., 2009).

Metinden konuşma sentezleme sistemindeki başka bir problem de tonlama ile ilgilidir. Tonlamanın nerede ve nasıl yapılması gerektiği, içeriği ve anlamı değiştirebilir. Çünkü tonun kullanımı kişiden kişiye ve insanların ruhsal durumlarına göre değişiklik gösterebilir. Aynı zamanda ulusların karakteristik durumlarına uygun şekilde farklı biçimler alabilir (Ergenç, 2002). Mesela İtalyanca ve İspanyolcada baskın olarak yüksek tonlu kullanım yaygın iken, Türkçede zayıf ton, İngilizcede ise çok zayıf ton kullanımı söz konusudur (Lemmetty, 1999)

Metin işleme bölümü tonlamayla da ilgilendiğinden doğal bir konuşma sentezleme için nerede tonlama yapılacağına karar vermek önemlidir. Bu işlem konuşma sentezi bölümünün sorumluluğundadır. Farklı tonlama modelleri olsa da bunlardan hiçbiri her durum için doğru bir şekilde çalışma başarısını gösteremeyebilir (Eker, 2002).

Konuşma sentezleme bölümü tarafından karar verilen doğru telaffuz ve tonlamalar tanınabilmelidir. Algısal olarak aynı sesler farklı metin içeriklerinde akustik olarak farklı olabileceğinden, bunu tanıyabilmek oldukça zor bir işlemdir. Çünkü MKS sistemi yalın bir metni ele aldığından kısaltma, sayı, noktalama işaretlerinde vurgu ve tonlamalar farklı telaffuz hatalarına sebep olabilmektedir.

### 2.3.5 MKS Sisteminde Doğallık ve Doğallığa Etki Eden Faktörler

Ses sentezleyiciler genellikle yapay (robotik) sesler üretirler. Bu mekanik veya robotik ses insan sesinden kolaylıkla ayırt edilebilir. Bazı şartlar altında bu robotik ses tercih edilebilir fakat çoğu zaman sentezleyiciden gelen sesin daha kolay anlaşılabilir ve dinlenebilir olabilmesi için insan sesine benzemesi tercih edilir. Konuşma sentezleyicilerin kalitesinin değerlendirmesi yapılırken iki önemli faktör üzerinde durulur. Bunlar anlaşılabilirlik ve doğallıktır. Anlaşılabilirlik sentezlenen konuşmanın kullanıcılar tarafından güvenli olarak anlaşılmasının göstergesidir. Doğallık ise sesin ne kadar insan sesine yaklaşılabildiğiyle ve kullanıcılara bir insanla konuşuyormuş hissi vermesiyle alakalıdır (Şirin Y. ve Kutlugün M.A., 2017).

Doğal bir konuşma sentezlemeyi engelleyen unsurlardan bazıları, konuşma parçalarının sürelerinin uygun şekilde modellenmesi ve uygun ezgi kurallarının tanımlanması olarak gösterilebilir (Uslu, 2010).

Bir cümleyi en doğru biçimde ifade edebilmek, uygun vurguyu yapabilmek, doğru ses perdesini ayarlayabilmek için cümlenin anlamına hâkim olmak şarttır. Henüz bu problemleri mükemmel bir şekilde çözebilecek bir sistem bulunmamaktadır.

### 2.3.6 Bazı Önemli MKS Uygulamaları

#### - MITALK

Program biçimlendirici (formant) türde sentezleme yapan, 1979 yılında MIT laboratuvarlarında geliştirilen bir uygulamadır. Günümüzde kullanılan birçok yeni uygulamanın temelini atmıştır (Allen, J. ve ark., 1987).

#### - INFOVOX

İsveç Royal Institute of Technology’de 1982 yılında geliştirilmiş olan Infovox ticari bir uygulamadır. MKS sistemlerinde oldukça yaygındır. İlk başlarda basamaklı biçimlendirici yöntemini kullanır iken, daha sonraki sürümlerinde çift-ses ekleme yöntemi kullanılmaya başlanmıştır. Farklı birden fazla dil için desteği mevcuttur. Erkek, bayan, çocuk gibi farklı seslendirmeler ile kaliteli ve anlaşılır konuşmalar elde edilebilmek mümkündür (Şentürk, 2010), (Ljungqvist M. ve ark., 1994).

#### - BELLABS TTS SYSTEM

Çift-ses ve üçlü ses ekleme yöntemleri ile birçok dil desteği vardır (Mönius B. ve ark., 1995). Modüler yapısı sayesinde hem ticari hem de gramer açısından oldukça olumlu sonuçlar elde edilebilmektedir (Şentürk, 2010).

#### - CNET PSOLA

Fransa Telekom tarafından 1980’lerde geliştirilen uygulama çift-ses ekleme yöntemi ile sentezleme yapmaktadır. Farklı birçok dil seçeneği bulunmaktadır. Seslerin birleştirilmesi için PSOLA algoritmasını kullanır (Şentürk, 2010).

#### - ETI ELEQUENCE

ABD’de geliştirilmiş bağlamalı bir sentezleyicidir. Çoklu dil desteğine sahip bir uygulamadır. Eklemeli yöntem kullanarak sentezleme işlemini gerçekleştirmektedir (Güldalı, 2009).

#### - FESTIVAL TTS SYSTEM

1990'lı yıllarda Edinburg Üniversitesinde bulunan Ses Teknolojileri Araştırma Merkezi tarafından geliştirilmiştir. Dilden bağımsız çalışabilir ve birçok farklı platformda hizmet verebilmektedir. İkili fonem birleştirme yöntemini kullanarak sentezleme yapar. Eğitim, araştırma ve bireysel kullanım için ücretsiz kaynak kodlu, açık sistem bir uygulamadır (Güldalı, 2009).

#### - MBROLA

Çoklu dil desteği bulunan bir konuşma sentezleyicisi olan MBROLA, konuşma modellerini dilbilim çalışmalarında test etmek için oldukça kullanışlıdır. Diphone birleştirici sentezleyici süre ve ses perdesi değerlerinin kolaylıkla kontrol edilmesine müsaade etmez. Fakat deneysel çalışmada kullanmak üzere yeni sentetik sesler oluşturmak daha kolaydır. MBROLA uygulamasında seslerinin belirli bir dilde tüm seslere sahip olmasına gerek yoktur. Bazı görevler için sadece birkaç difondan mikro sesler çıkarmak yeterli görülür. Son zamanlarda, geliştirilmiş konuşmalar elde edilebilmek için yeni MBROLA mikro sesleri geliştirilmeye başlanmıştır (Bachan, 2010).

#### - MICROSOFT

Microsoft Konuşma API'si (SAPI), Microsoft tarafından geliştirilen bir uygulamadır. Windows uygulamaları üzerinde konuşma tanıma ve konuşma sentezleme kullanmaya olanak tanır. SDK, Windows işletim sisteminin kendisine entegre edilmiştir. SAPI kullanan uygulamalar arasında Microsoft Office ve Microsoft konuşma sunucusu bulunmaktadır. Konuşma API'si serbestçe dağıtılır ve konuşma teknolojisi kullanan tüm Windows uygulamaları ile birlikte gönderilebilir.

#### - GOOGLE TTS

Google text-to-speech sistemi, Android işletim sistemi için geliştirilmiş bir uygulamadır. Uygulama çoklu dil desteğine sahiptir. Google Çeviri gibi Google hizmetlerini kullanır. Hizmet, Android platformunda geliştiricilere API'ler sunar. Google Translate hizmetini kullanarak API'lere resmi olmayan erişim imkanı da vardır. Google TTS, tarayıcıdaki herhangi bir metni okumak için Chrome web tarayıcısında uygulanmıştır (Gebremariam, 2016).

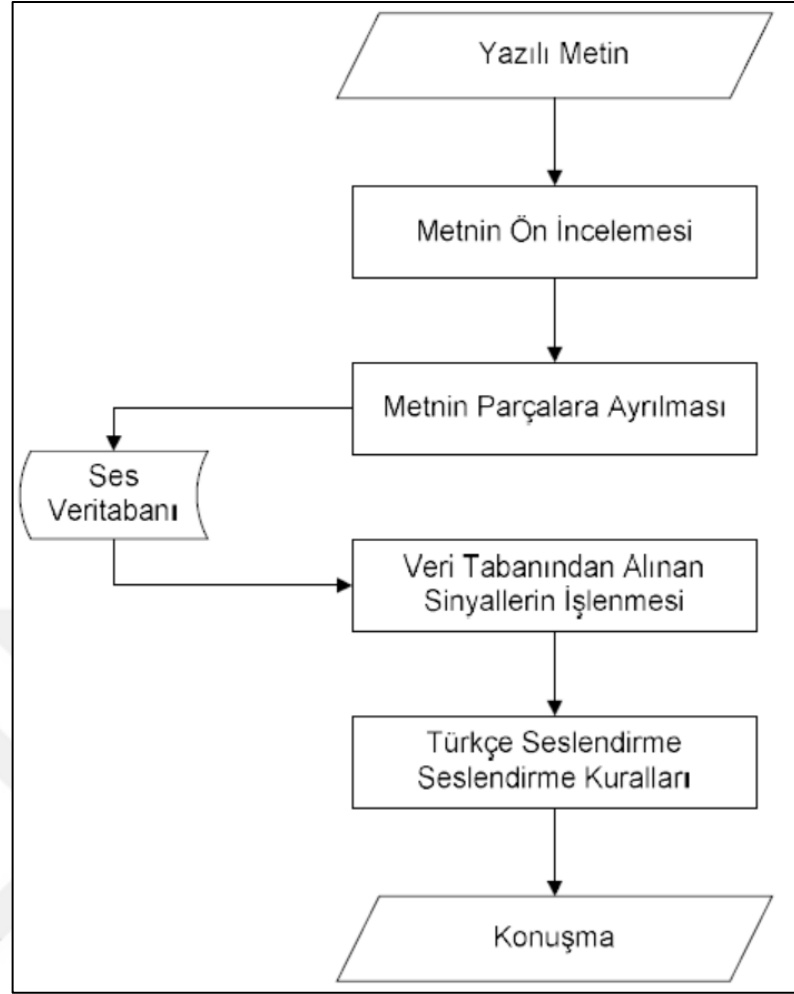
## - SESTEK / GVZ

Ses ve iletişim teknolojileri, çağrı merkezi uygulamaları ve müşteri hizmetleri otomasyonu alanlarında faaliyet gösteren SESTEK firması 2000 yılında kurulmuş, 2007 yılında GVZ Ses Tanıma Teknolojileri şirketini bünyesine katmıştır. SESTEK konuşma sentezi (Text-to-Speech / TTS) yazılımı, elektronik ortamdaki metinlerin anlaşılabilir bir biçimde ve insan sesi doğallığında seslendirilmesini sağlamaya çalışan bir text-to-speech uygulamasıdır. Farklı dil seçenekleri ve spiker sesleri ile çözüm alternatifleri sunabilmektedir. SESTEK adres, isim-soyisim, kredi ekstresi, doğum yeri gibi veri tabanındaki müşteri bilgilerini okuyabilmekte ve işlem teyidini gerçekleştirebilmektedir.

### **2.4 Türkçe Metinden Konuşma Sentezleme Sistemleri**

Dünya genelinde mevcut batı dillerinde geliştirilen sözcük tabanlı çok sayıda konuşma sentezleme yazılımı vardır. Bu yazılımlar üç farklı MKS metodu ile sentezleme işlemini gerçekleştirmektedirler. Bu metotlar; söyleyiş veya boğumlama (articulatory) sentezleme, biçimsel (format) veya kural tabanlı sentezleme ve eklemeli (concatenative) sentezleme şeklindedir. Türkçe Latin kökenli batı dillerine göre sondan eklemeli bir dil olması sebebiyle önemli farklılıklara sahiptir (Aksan D., 2000). Bu sebeple çoğu uluslararası üretilen yazılımlarda dil desteği sunulmamaktadır. Ancak Türkçenin yazıldığı gibi okunan bir dil olması sebebi ile tüm bunlara rağmen az sayıda da olsa bazı yazılımlarda sentezlenen Türkçe metinlerin anlaşılır olduğu görülebilmektedir. Bu istisnai durum haricinde Türkçenin kendine özel yapısından dolayı daha farklı metotlar ile ele alınması gerekmektedir (Tekindal B. ve Arık G., 2012).

Türkçe, morfolojik açıdan eklemeli bir dil olduğundan, bir sözcükten daha başka birçok sözcük türetilerek elde edilebilmektedir. Bundan dolayı dildeki sözcük sayısı daha kolay çoğaltılabilmektedir (Sel İ. ve ark., 2011). Aşlıyan ve arkadaşları bu sebeple Türkçe dili için geliştirilecek MKS sistemlerinde kullanılması gereken en uygun metodun eklemeli sentezleme metodu olduğunu vurgulamaktadırlar (Aşlıyan R. ve ark., 2006). Türkçe için eklemeli sentezleme metodu kullanılarak oluşturulan bir örnek model aşağıda gösterilmektedir.



Şekil 2.6. Eklemeli sentezleme sistemlerinde bulunan temel adımlar (Sel, 2013)

Türkçe MKS sistemleri dört ana aşama ile ele alınmaktadır. Bunlar;

- Metin ön işlemlerinin yapılması
- Metnin hecelere ayrılması
- Ses veri tabanının oluşturulması ve
- Seslerin birleştirilerek seslendirmenin gerçekleştirilmesi aşamalarıdır.

Bu aşamaları açıklamaya geçmeden önce ses, fonem ve hece gibi bilinmesi gereken bazı kavramlar vardır. Bu kavramlar aşağıda açıklanmaktadır.

İnsan düşünce olarak beyinde oluşturduğu kavramları konuşarak dile dökmek istediği zaman, konuşacağı dildeki en uygun kelimeleri ve aynı zamanda konuştuğu dilin yapısına uygun kelime dizilerini belirlemektedir. Bundan sonra ses üretimi konuşma organlarının harekete geçmesi ile gerçekleşmektedir (Erdemir C., 2010).

Bir dilin başlıca elemanı ses olarak tanımlanmaktadır. Seslerin yazılı ifadesi olarak harflerden yararlanır. Harfler yazılı, ses ise sözeldir. Sesler, söz şeklinde ağızdan çıkan dile ait parçalardır. Bunlar yazıya çevrilirken harfler kullanılır ve bir dilde kullanılan harf çeşitlerinin tamamı alfabeyi meydana getirir. Türkçe alfabede 29 adet harf vardır. Bunlar genel olarak ünlü ve ünsüz harfler şeklinde iki gruba ayrılmaktadır (Şentürk, 2010). Bir cümle kelimelerden, kelimeler hecelerden, heceler de fonemlerden (sesbirimlerden) meydana gelir. Ses, diller üstü bir birim iken fonem ise anlam ayırıcı özellik içermesi bakımından belli bir dile özgüdür (Artuner, 1994; Can, 2007).

Fonem, ünlü ve ünsüz ses kümelerinden oluşur. Fonemler, anlam ayırıcı özelliğe sahiptir. Anlam ayırıcı özelliği bulunmayanlar tek bir fonem kümesi altında toplanmaktadır (Artuner, 1994; Can, 2007). Kısaca fonem, belli bir dilde bir kelimededen diğerine farklılık gösteren en küçük ses birimidir. Eğer herhangi bir sesli ifadede bir ses biriminin değiştirilmesi onun anlamını da değiştiriyor ise, bu durumda değiştirilen birim aynı zamanda fonemdir (Can, 2007).

Fonemler birleşerek heceler oluşur. Tüm dillerde bu uygulanmaktadır. Fakat heceler farklı dillerde farklı biçimlerde olduğundan heceleri tanımlamak daha zordur. Her dil için heceler, yapıları bakımından farklılık göstermektedir. Bu yüzden tüm dilleri kapsayacak ortak bir hece tanımı yapmak mümkün değildir (Şentürk, 2010). Tam olarak bir tanımı olmamasına rağmen heceler, seslendirilebilecek şekilde oluşturulan bir ünlü ve birkaç ünsüz harfin birleşimi ile meydana gelen harf dizisi şeklinde tanımlanabilir (Erdemir C., 2010).

#### 2.4.1 Metin Önışleme Aşaması

Metin önışleme süreci karmaşık ve her dil için özel çözümler gerektiren bir süreçtir (Güldalı, 2009). MKS sistemleri normalde verilen herhangi bir metni doğru biçimde seslendirebilmelidir. Yani sisteme girilen metin içerisinde harfsel ifadelerin yanında değişik biçimlerde ifadeler de bulunmaktadır ve bunlar seslendirilme sürecinde dile uygun olarak ele alınmalıdır. Bunun için metin önışleme sürecine ihtiyaç vardır (Ünalı, 2007).

Metin önışleme aşaması kendi içinde ayrı bazı işlemler içermektedir. Metin önışleme aşamasında rakamlar, sayılar, kesirli ifadeler, tarihsel biçimler, sıralama ifadeleri,

kısaltmalar ve özel bazı karakterler gibi yazı dilinde okumayı anlamlı kılan ifadeler, uygun sözcük gruplarına dönüştürülmelidir.

Türkçede sayı değerlerinin seslendirilmesi için yazının okunabilir yazı dizisi haline getirilmesi doğru seslendirme için önemlidir. Tamsayılar, ondalıklı ifadeler, tarihler ve saatler, telefon numaraları ve T.C. kimlik numarası gibi özel biçimleri olan her sayısal ifade farklı bir şekilde seslendirilmelidir. Örnek olarak; “571” sayısı “beşyüzyetmişbir” gibi yazı ile ifade edilecek bir biçime dönüştürülmelidir. Böylece rakamsal ifadeleri önce yazı haline dönüştürmek daha doğru biçimde seslendirme sağlayacaktır. Ancak bazı marka-model bilgisi içeren yazım şekilleri özel seslendirmelere ihtiyaç duyabilir.

Kesirli ifadeler de Türkçe dilinde kendilerine özgü şekilde okunmalıdır. Örneğin; “2/5” kesir ifadesi “2 bölü 5” şeklinde okunabilir. Tarihsel “01.01.1991” veya “01/01/1991” şeklinde yazılan ifadeler “bir ocak bindokuzyüzdoksanbir” şeklinde çözümlenebilmektedir.

11 hane içeren cep telefonu numaraları 05351231213: 0(sıfır) – 535(Beşyüzotuzbeş) – 123(yüzyirmiyüç) – 12(oniki) – 13(onüç) şeklinde Türkçe olarak seslendirilmektedir.

11 haneli olan 12345678911 T.C. kimlik numarası 123(yüzyirmiyüç) – 456(dörtüyzellialtı) – 789(yediyüzseksendokuz) – 11(onbir) şeklinde seslendirilebilmektedir.

Bir kelime, terim veya özel ismin içindeki bazı harflerden yararlanarak oluşturulan daha kısa biçimli simgeleştirilmiş haline kısaltma denir (Canal Ş.M. ve ark., 2010). Kısaltmaların okunmasına da dikkat edilmelidir. Kısaltmalar, harf harf okunabildiği gibi, bir kelime gibi de okunabilir ya da tümüyle başka bir şekilde okunabilir (Ünaldı, 2007). Örneğin; Türkçede “PTT” kısaltması “Pe Te Te” ya da “Posta Telefon Telgraf” şeklinde seslendirilebilmektedir (Vural, 2003). İlk önce kısaltmaların tespiti yapılmalı, bazı istisnai durumlar için ayrı bir sözlük oluşturulmalıdır.

Yazım kuralları çerçevesinde küçük harfli kısaltma içeren ifadeler sözcüğün okunuşuna göre ek alırlar. Örneğin kg.’ında → kilogramında vb. Büyük harfli

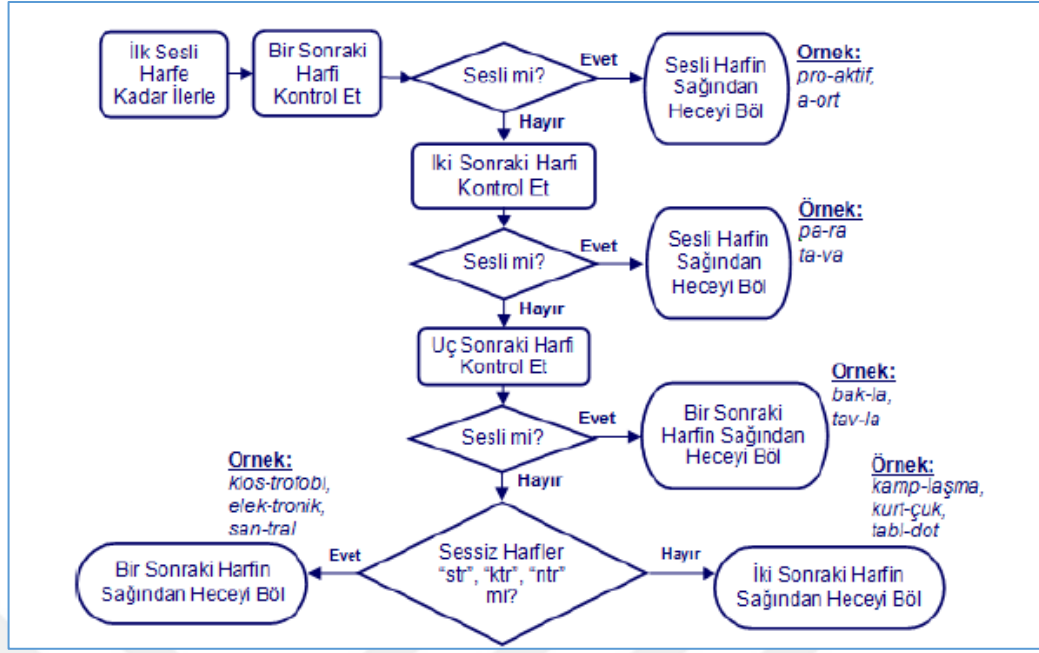


kısaltmalar ise kısaltmanın okunuşuna göre ek alırlar. Örneğin TBMM → tēbēmēmē (Yılmaz A. E., 2009).

Bunların yanında noktalama işaretleri de metnin telaffuzuna etki eden, vurgu ve tonlama açısından önemli bir etkidir. Özel işaretler ise belirli okunuşlara sahip olmakla birlikte farklı okudukları durumlar da görülebilmektedir. Örnek olarak; “artı” ve “eksi” şeklinde karşımıza çıkan “+” ve “-” işaretleri bir sayının önüne geldiğinde “artı” olarak okunup kan grubu olarak ifade edilirken “pozitif” ve “negatif” olarak da yorumlanabileceği kullanımlar mevcuttur (Canal Ş.M. ve ark., 2010). Ayrıca diğer noktalama işaretlerinden “virgül (,)” için yarım ölçü ve “nokta (.)” için bir tam ölçü duraklama eklenmektedir. Bunların dışında kalan ve metin içerisinde geçen “/, \$, %” vb. işaretlerin de doğru bir biçimde yorumlanması ve seslendirilmesi gerekmektedir. (Yılmaz A. E., 2009).

#### 2.4.2 Metnin Hecelere Ayrılması Aşaması

Türkçe eklemeli bir dil olduğundan bir sözcükten, ek getirmek suretiyle birçok kelime türetilmektedir. Bu yüzden Türkçede sözcükleri ses birimi olarak kullanmak mümkün olmamaktadır. Bunun yerine ses birimlerini elde etmek için hecelerin kullanılması daha uygun görülmektedir. Türkçe heceleme sistemi basit ve mekanik bir yapıya sahiptir. Bazı genel kurallar uygulanarak tasarlanacak bir algoritma ile ifade edilebilmektedir. Bunun için Aşlıyan ve arkadaşları TASA adı verilen Türkçe metinler için hecelere ayırma algoritması geliştirmişlerdir (Aşlıyan, R., ve Günel, K., 2005). Metni hecelere ayırmak için kullanılan akış diyagramı aşağıdaki şekilde gibidir.

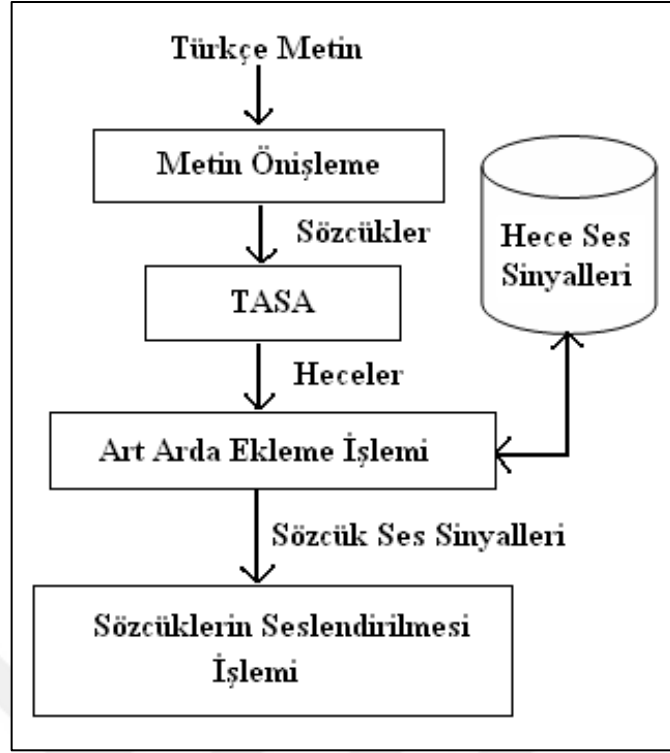


Şekil 2.7. Heceleme algoritması (Aşlıyan, R., ve Günel, K., 2005).

#### 2.4.3 Ses Veri Tabanının Oluşturulması Aşaması

Girdi olarak verilen metindeki cümleler kelimelere, kelimeler de ses parçacıklarına ayrıldıktan sonra, veri tabanından uygun ses kaydının bulunması aşaması başlamaktadır. Hecelere ayrılan metinsel ifade, önceden seslendirilmiş olan ve ses veri tabanına etiketlenmiş olarak kaydedilen ses verileri ile eşleştirme yapılarak sentezlenme işlemi tamamlanmaktadır (Tekindal B. ve Arık G., 2012). Etiketleme, elde edilen ses birimlerinin bir veri tabanında isimlendirilerek kaydedilmesidir. Sistemin çalışması gerekli olan ses birimlerinin daha önce kayıt edilmiş olmasına bağlıdır.

Türkçe alfabeti incelendiğinde kaydetme işlemi için üretilmesi gereken sekiz farklı hece türü bulunmaktadır. Bu heceler en az bir harf, en fazla dört harften meydana gelmektedir. Aslında Türkçe'nin özünde altı farklı hece tipi vardır, ancak günümüzde diğer başka dillerden Türkçeye dâhil olan ve dilimizce kabul edilmiş birçok sözcük bulunmaktadır. Bu hece tipleri dikkate alınmadığında çoğu metin seslendirmede problem meydana gelmektedir (Tekindal B. ve Arık G., 2012), (Şentürk T. ve Adalı E., 2010). Sistemin genel yapısını gösteren özet bir gösterim aşağıda sunulmuştur.



Şekil 2.8. MKS sisteminin genel süreci (Aşlıyan R. ve Günel K., 2008)

Sistem özet olarak, metin içerisindeki sözcüklerin girdi olarak alınmasıyla başlamakta ve bu girdiler TASA'ya gönderilerek Türkçe heceler veri tabanı oluşturulmaktadır. Bundan sonra heceler için önceden kayıtlı bulunan ses veri tabanından ilgili ses birimleri seçilerek uygun şekilde art arda ekleme işlemi ile seslendirme yapılmaktadır (Aşlıyan, R., ve Günel, K., 2005).

Ses birimi için ikili fonem tercih edildiğinde buna uygun bütün ikili fonemlerin oluşturulması, hece tabanlı bir sistem seçildiğinde ise Türkçe için önceden belirlenen hece listesinin tümünün kayıt altına alınması gerekmektedir (Güldalı, 2009). Ancak seslerin tamamının tek tek kaydedilmesi ve etiketlenmesi aşırı zaman ve efor harcanmasını gerektirmektedir. Bu sebeple bu seslerin dış kaynaklardan temin edilmesi yönünde çalışmalar yapılmaktadır (Şentürk, 2010).

#### 2.4.4 Seslerin Birleştirilmesi ve Seslendirme Aşaması

Düzgün bir seslendirme yapılabilmesi için ses parçacıklarının seslendirilme süreleri ve sese ait bazı akustik özellikler gibi sinyal işleme işlemlerinin doğru bir biçimde belirlenmesi gerekmektedir. Metin içinde geçen her parçacığın (fonem) konumunun

da dikkate alınması gerekmektedir. Aynı zamanda düzgün bir seslendirme için parçacık süreleri çok iyi belirlenmelidir (Erdemir C., 2010).

Ses dalgası, eşleşen fonemlere ait ses birimleri ve durak noktalarının birleştirilmesi ile üretilir. Ses dalgası üretmenin daha başka birçok yolu vardır. Mevcut sistemler genel olarak iki yol ile bunu gerçekleştirirler. Bunlardan biri kayıtlı bulunan hazır insan sesi parçalarının birleştirilmesi, diğeri ise sinyal işleme tekniklerinden yararlanılmasıdır (Tekindal B. ve Arık G., 2012).

Ses kütüphanesi rasgele oluşturulmuş ve belirli kurallara uyulmamış ise, buna çözüm olarak mutlak değer veya enerji gibi bazı üstel fonksiyonlar kullanmak veya sıfırı kesen nokta sayısı vb. başka hesaplamalar yapmak suretiyle, ünlü ses sürelerinin algoritmik olarak tespit edilmesi gerekmektedir (Erdemir C., 2010). Eklemeli MKS sistemlerinde daha önceden kaydedilen farklı tonlama içeren seslerin birbirlerine istenilen ton ve sürelerde birleştirilmesi ve bir akışkanlık elde edebilebilmesi için zaman ölçeği modifikasyonları kullanılmaktadır. Genel olarak SOLA, TD-PSOLA ve WSOLA şeklinde tanımlanan modifikasyon teknikleri mevcuttur (Sel, 2013).

Birleştirerek sentezleme işlemi yapan MKS sistemlerinde işlem yükü diğeri sentezleyicilere göre daha azdır ve bunun yanında konuşmanın doğallığı daha fazladır (Özen, 2002). Perde frekansı, enerji seviyesi ve fonem uzunluğu gibi konuşma için önemli olan unsurlar, konuşma veri tabanındaki parçaların farklı metinlerden elde edilmesinden dolayı farklılık gösterecektir. Doğru bir sentezleme yapılabilmesi için bu farklı özelliklerin kabul edilebilir uygun biçimler haline getirilmesi önemlidir (Ünaldı, 2007).

Konuşmada ses birimi için en küçük birim fonemdir. Bundan dolayı oluşturulabilecek en küçük konuşma veri tabanı fonemlerden elde edilir. Türkçede 43 adet fonem vardır. Ancak fonem kullanılarak yapılan birleştirme ile seslerin oluşturulması MKS sistemlerinde kullanılmamaktadır. Bu durum, fonemlerin birleşme noktalarının belirli olmayışı ve tek tek kaydetme işleminin problemlere yol açmasından kaynaklanmaktadır (Üçok, 1951). Kendinden önce ve sonra gelen fonemler birbirlerine etki ettiği için tek başlarına kayıt edilmesi yerine bitişik fonemler şeklinde toplu olarak kaydedilip kullanılmaktadırlar (Ayhan, 1998).

Difon kullanımı daha yaygın olan bir yöntemdir. Difonlar, iki tane yarım fonemden oluşan ses biçimleri olarak adlandırılırlar. Bunlar fonemlerden daha hassas birleşme bölgelerine sahip şekilde veri tabanında kayıtlı bulunurlar. Difonların sınırları, fonemlerin daha durağan yani işlenmesi daha kolay merkez kısımları olarak tayin edilmektedir. Bunlardan daha da küçük parça boyutlarının kullanılması, daha fazla işlem gerektirmesi ve birleşme noktalarında da daha çok bozulmaya sebep olması anlamına gelir. Daha az işlem yapılması ve daha az bozulma elde edilmesi için daha büyük konuşma parçaları kullanılmalıdır (Ünaldı, 2007).



## 3 MAKİNE ÖĞRENMESİ

### 3.1 Makine Öğrenmesi Nedir?

Makine öğrenmesi, bilgisayarın meydana gelen bir olay ile ilgili edindiği bilgileri ve tecrübeleri öğrenmek suretiyle, gelecekte oluşabilecek benzer olaylar hakkında kararlar verebilmesi ve oluşacak problemlere çözüm üretebilmesidir (Öztemel E., 2006). Makine öğrenmesi bazı yöntemler kullanarak geçmişteki verilerden yararlanır ve yeni veri için en uygun modeli bulmaya çalışır.

Çok büyük miktarlardaki verinin elle işlenmesi ve analizinin yapılması oldukça zordur. Burada amaç geçmişteki verileri kullanarak gelecek durumlar için tahminlerde bulunabilmektir. Uygulama alanı ne olursa olsun, çok miktardaki verinin analiz edilerek gelecek ile ilgili tahminlerde bulunması ve bizim karar vermemize yardımcı olması sayesinde makine öğrenmesi yöntemlerinin önemi her geçen gün artmaktadır.

Yapay Zekânın alanının bir dalı olan Makine Öğrenmesi (Machine Learning), bilgisayarların “öğrenme” görevini yerine getirecek algoritma ve tekniklerin gelişimi ile ilgilenir. Makine öğrenmesi; Doğal Dil İşleme, Konuşma ve El Yazısı Tanıma, Nesne Tanıma, Bilgisayar Oyunları, Robot Hareketleri, Arama Motorları ve Tıbbi Teşhis gibi birçok alanda kullanılır (Kutlugün ve ark., 2017).

Makine öğrenmesinin üç önemli aşaması vardır. Bunlar:

1. Dokümanların Hazır Hale Getirilmesi
2. Öğrenme Yöntemlerinin Kararlaştırılması ve Uygulanması
3. Öğrenme Performansının Değerlendirilmesi

Makine öğrenmesinde öncelikle öğrenme işleminin yapılacağı veri kümesinin uygulanacak öğrenme yöntemine uygun halde hazırlanması gerekmektedir. Öğrenme metodunda istatistiksel yöntemler kullanılmaktadır. Yeni geliştirilen metotlar da

istatistiksel temellidir. Yeni bir metot bulunduğunda bu metodun performansı ölçülmektedir. Bu sayede diğer metotlarla karşılaştırılması yapılmaktadır (Uzun, 2007). Makine öğrenmesi metotlarını farklı uygulamaları analiz etmek ve farklı beklentilere göre sınıflandırmak mümkündür (Alpaydın, 2004).

Makine öğrenmesi yöntemleri aşağıda açıklamaları verilen işlemleri yerine getirmek üzere kullanılabilir.

- Sınıflandırma (classification): Sınıfı belli olan verilerden yararlanarak yeni bir durumun sınıfının doğru tahmin edilebilmesi işlemidir.
- Kümeleme: Önceki bilgilerin bilinmediği durumlarda verilerin birbirine benzer olma durumlarına göre elde edilen kümelerin belirlenmesi yani, bir kümeyle ait olması işlemidir.
- Eğri Uydurma (Regresyon): Önceki verilere karşılık gelen sınıflardan ziyade, sürekli değerlerin yer aldığı problemlerdir.
- Özellik seçimi ve çıkarımı: Veriye ait bazı özelliklerden yararlanarak verinin kümesinin veya sınıfının karşılığını belirleyen özelliklerin tespit edilmesi işlemidir.
- İlişki belirleme: Elde edilen sınıf veya kümeler arasındaki ilişkilerin çözümlemesi işlemidir.

Makine öğrenmesi teknikleri gözetimli ve gözetimsiz öğrenme metotları olmak üzere ikiye ayrılır.

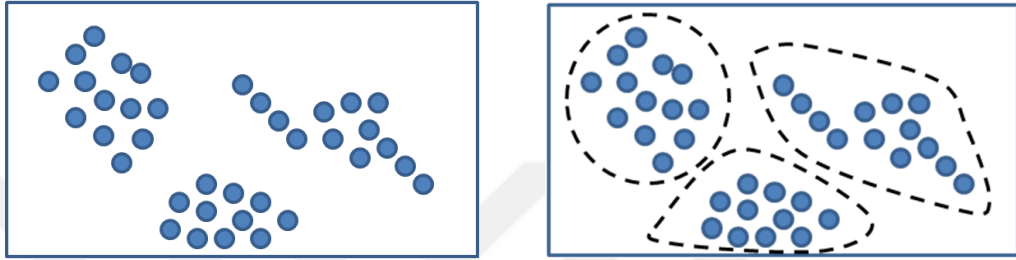
### 3.1.1 Gözetimsiz Öğrenme

Gözetimsiz öğrenme, etiketlenmemiş ham veri kümesindeki veriler arasında var olan ancak açıkça görülmeyen bağıntının açığa çıkarılması işlemidir (Nizam, H. ve ark., 2014). Gözetimsiz öğrenme (Unsupervised Learning) gözlemlere bağlı bir makine öğrenmesi yöntemidir. Diğer bir deyişle çıktı verilerini kullanmadan sadece girdiler üzerinde öğrenme işlemini gerçekleştirmeye çalışır (Hinton G. ve Sejnowski T.J., 1999). Gözetimsiz öğrenme tekniği, veri sıkıştırma işlemlerinde oldukça kullanışlıdır.

Gözetimsiz öğrenmede amaç tanıma veya sınıflandırma değildir. Çünkü veri kümesindeki örneklerin çıkışları bilinmemektedir. Genellikle kümeleme, olasılık

yoğunluk tahmini, öznitelikler arasındaki ilişkilerin bulunması ve boyut indirgeme gibi amaçlar için kullanılmaktadır (Chao, 2011).

Gözetimsiz öğrenme tekniği özellikle olasılık teorisine dayanan kümeleme işlemlerinde kullanılmaktadır. Kümeleme, farklı gruptaki birbirine benzer olan nesnelerin sınıflandırılmasıdır. Kümeleme, makine öğrenmesinin yanında veri madenciliği, örüntü tanıma ve resim analizi gibi birçok alanda da kullanılmakta olan istatistiksel veri analizini gerçekleştirmek için genel bir tekniktir (Uzun, 2007).



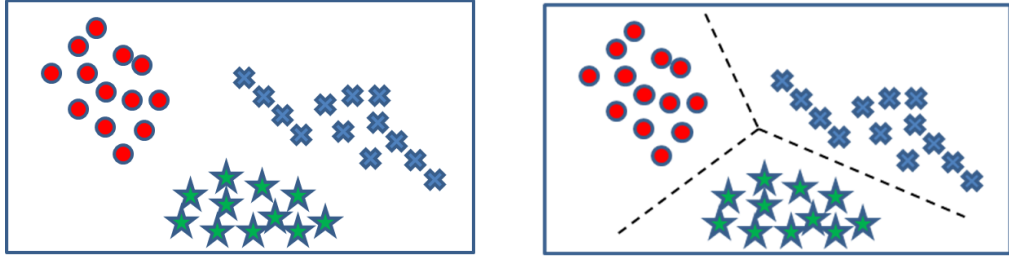
Şekil 3.1. Gözetimsiz öğrenme

### 3.1.2 Gözetimli Öğrenme

Gözetimli öğrenme, önceden gözlenerek sonuçları bilinen (etiketlenmiş) verilerden yararlanarak bu verilerin ve sonuçların tümünü kapsayan bir fonksiyon elde etmeyi amaçlayan makine öğrenmesi yöntemidir (Nizam, H. ve ark., 2014). Kısaca gözetimli öğrenme (Supervised Learning) eğitim verileri üzerinden bir fonksiyon üreten tekniktir. Bu öğrenme tekniğinin algoritması girdilerle (etiketlenmemiş veri) çıktılar (etiketlenmiş veri) arasında en iyi eşleme yapan fonksiyonu üretmek üzere çalışmaktadır. Eğitim verisi, girdi nesnelere ve sonuçlanması istenilen çıktılardan meydana gelmektedir.

Sistem eğitilirken veri setinde bulunan her bir örneğe ait giriş ve çıkışlar verilir. Sınıflandırma çalışmalarında metnin içeriği giriş verisi olarak kullanılırken, çıkış için ise metne ait kategori kullanılmaktadır. Sistemin doğrulanması amacıyla test veri setinden yararlanır. Doğrulama aşamasında öğrenme algoritması kategorisini bilmediği bir test verisine, eğitim verisinde bulunan çıkışlardan uygun olan birini atar (Kotsiantis, S. B., 2011).





Şekil 3.2. Gözetimli öğrenme

### 3.2 Sınıflandırma

İnsan beyni yüz tanıma, ses tanıma gibi her gün hayatın her safhasında sınıflandırma yapmaktadır. Sınıflandırma, farklı bilgi parçalarını tanıma işlemidir. Örüntü tanımda eldeki bilgilere göre kararlar almaya veya tahminler yapmaya yarar. Sınıflandırma ile toplanan veriler kullanım amacına göre farklı kategorilere ayrılarak modellenir ve sistem tanımlanır. Bu sistemler daha sonra, konuşma tanıma, parmak izi tanıma, DNA dizilimlerini doğrulama gibi farklı uygulamalar şeklinde karşımıza çıkar (Therrien, 1989).

Başlangıçta farklı sınıflara ait nesnelere verilerek daha sonra karşılaşılan sınıfı bilinmeyen yeni bir nesneyi bu sınıflardan birine atama problemine sınıflandırma problemi adı verilir (Yılmaz R. , 2013). Verinin incelenip, içerisinden işe yarayan bilginin çıkarılmasına da Veri Madenciliği (data mining) adı verilir. Özellikle doğal dil tanıma probleminde en sık kullanılan makine öğrenmesi tekniği sınıflandırmadır ve metin sınıflandırma bu alanda önemli bir yer tutar.

Sınıflandırma, gözetimli öğrenmenin bir çeşididir. Özetle, veriyi doğal bir benzerlik veya uzaklık ölçüsüne dayalı olarak kategorilere ayırmayı içerir.

#### 3.2.1 Bazı Önemli Kavramlar

**Eğitim Kümesi (Örnekleme Kümesi)(Training Set):** Önceden karşılaşılmış veya elde edilmiş girdiler ve bu girdilere ait kategorisi bilinen çıktılar kümesinin bütününe verilen isimdir. Bu küme yardımıyla karşılaşılabilecek yeni örneğin sınıfı belirlenmeye çalışılır.

**Gürültü (Noise):** Konu dışı ve anlamsız bilgi demektir. Makine öğrenmesi algoritmalarının hedefi gürültülü veriyi mümkün oranda azaltmak ve doğru sınıflar

belirlemeye çalışmaktır. Gürültü, eğitim verisi için de çok önemlidir. Eğitim kümesinde gürültülü veri bulunması durumunda, yeni gelen eğitim verilerinin tahmini zorlaşmaktadır.

Çapraz Doğrulama (Cross Validation): Eğitim verisi alt kümelerle ayrılarak bir kısmı eğitim için kullanılırken diğer kalan kısmı doğrulama işlemi için kullanılır. Bu işlem çapraz bir şekilde tüm alt kümeler için tekrar edilir. Bu işleme çapraz doğrulama adı verilir (Uzun, 2007).

Veri kümesi 'n' adet eşit parçalara ayrılır ve ilk grup test kümesi, diğerleri eğitim kümesi olarak atanarak sınıflandırma yapılır. Daha sonraki adımlarda izleyen alt grup test kümesi, geri kalan tüm gruplar eğitim kümesi seçilir. Bu işlemler n kere her alt grup için aynı şekilde tekrar eder. Sistemin genel başarımı bu her sınıflandırma sonrası elde edilen tüm sonuçların ortalaması alınarak hesaplanmaktadır. Literatürde en sık kullanılan 'n' değeri 10'dur (Pehlivan, 2014).

10 kat çapraz doğrulama için veri kümesi 10 eşit parçaya ayrılır. Her bir işlem adımında veri kümesinin 9/10'luk bölümü eğitim kümesi, 1/10'luk bölümü ise test kümesi olmak üzere kullanılır. Her adımda farklı bir parça test kümesi olacak, diğer kalan parçalar eğitim kümesi olacak şekilde işlem sürdürülür. 10 kere çapraz doğrulamanın genel yapısı aşağıdaki şekilde verilmiştir (Biricik, 2011).

	1. Parça	2. Parça	3. Parça	4. Parça	5. Parça	6. Parça	7. Parça	8. Parça	9. Parça	10. Parça
1. Adım	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
2. Adım	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
3. Adım	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
4. Adım	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
5. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
6. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim
7. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim
8. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim
9. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim
10. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test

Şekil 3.3. 10-Kat çapraz doğrulama

10 kere çapraz doğrulama ile veri kümesindeki tüm örnekler eğitim ve test kümesi olduğu için önemli bir avantaj sağlanmaktadır (Biricik, 2011).

### 3.3 Metinler Üzerinde İşlemler

Bilinmesi gereken kavramlardan sonra tez konumuzu içeren metinden konuşma sentezleme amacına makine öğrenmesi tekniklerini uygulayabilmek için metin dokümanları bazı hazırlık aşamalarından geçirilmektedir.

Metinsel veriler üzerinde bazı analiz ve çıkarımların yapılabilmesi için ilk olarak bu veriler üzerinde ön işlemlerin yapılması gerekmektedir. Bu ön işlemleri uygularken metin madenciliğinde önemli olan bazı tekniklerden faydalanılmaktadır. Bunlar veri temizlemenin yanında veriyi uygun bir biçime getirme işlemini de içer (Fieldman, R. and Sanger J. , 2006)]. Metinsel veriler yapısal hale getirilerek kullanıma sunulur.

Metin içeren dokümanlara ön hazırlık aşaması için 5 farklı işlem uygulanmaktadır. Bunlar;

- Dokümanların Ayrıştırılması
- Gereksiz Kelimelerin Temizlenmesi
- Kelime Köklerinin Tespiti
- Terim Ağırlıklandırma
- Boyutsal İndirgemedir.

#### 3.3.1 Metin Önişleme

Metin içeren dokümanları makine öğrenmesi teknikleri kullanarak sınıflandırmak veya kümelere ayırmak için önce dokümanların bir ayrıştırma işleminden geçirilmesi gerekmektedir. Örneğin düz bir metinden ses dosyaları elde edebilmek için metnin doğru metotlarla işlenmesi ve yorumlanması gerekmektedir. Bunun için metinler önişleme sürecinden geçirilir ve bazı hesaplamalar yapılabilmesi için sayısal değerlere dönüştürülür (Şirin Y. ve Kutlugün M.A., 2017). Bu dönüşümden sonra metin artık vektörel değerler olarak işlem görerek makine öğrenmesi algoritmaları uygulanabilmektedir.

Ön işlemede ilk olarak metin içermeyen noktalama işaretleri, boşluk ve sayısal değerler gibi, ‘stop words’ olarak adlandırılan, anlam için direk etkisi olmayan

karakterleri ayıklama işlemi yapılır. Daha sonra doküman içindeki kelimeler gruplara ayrılmaktadır.

Özetle metin önışleme süreci, dokümanların makine öğrenmesi teknikleri için elverişli duruma getirilmesi işlemidir. Bu durum için dokümanlar genellikle terim sayma modeli (term count model) ve vektör uzayı modeli (vector-space model) ile ele alınmaktadır (Özgür, 2004).

### 3.3.2 Terim Sayma Modeli

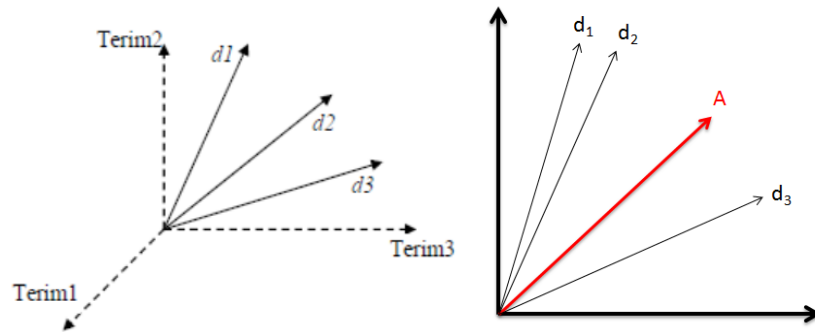
Terim sayma modeli basit bir modeldir. Bu model, indeksleme yapmak ve dokümanlar arası ilişki değerlendirmede kullanılmaktadır. Bu modelin dezavantajı fazla tekrar eden terimlerin ön plana çıkmasıdır (Uzun, 2007).

$$Genel\_Terim = \frac{D}{d(n)} \quad (3.1)$$

$D$ : Bir veri setindeki toplam doküman sayısı,  $d(n)$ : Terimin içerdiği doküman sayısı

### 3.3.3 Vektör Uzayı Modeli

Dokümanlarda indeksleme yaparken yaygın olarak kullanılan bir modeldir (Salton G. ve ark., 1975). Vektör uzayı modeli bilgi erişimi, bilgi filtreleme, indeksleme ve dokümanlar arasında ilişki değerlendirmede kullanılan matematiksel bir modeldir. Bu modelde, her doküman  $d$  vektörü ile gösterilir. Vektör  $d$  içindeki her boyut dokümanların terim uzayında farklı bir terimini göstermektedir (Uzun, 2007).



Şekil 3.4. Vektör uzayı modeli (Uzun, 2007) ve (Pilavcılar, 2007).

Her bir terim, bir kelimeyi veya birden fazla kelimedenden oluşan kelimeler grubunu ifade edebilmektedir. Kelime grupları, istatistiksel yöntemlerle veya doğal dil işleme

yöntemleriyle belirlenebilmektedir. İstatistiksel yöntemlerde kelimelerin bir arada kullanılma sıklıklarından yararlanılarak kelime grupları üzerinde işlem yapılabilmektedir (Cohen W. W. ve Singer Y., 1996).

Oluşan her d vektörü arasında bir açı meydana gelir. Bu açı değeri hesaplanıp karşılaştırılarak vektör uzayındaki benzerlikler hakkında bilgi sağlanabilmektedir. Kümeleme veya sınıflandırma algoritmalarında iki doküman arasındaki benzerliğin belirlenebilmesi için kosinüs benzerliği formülü kullanılmaktadır. Bu formüldeki cos değerine göre benzerlik oranı hesaplanmaktadır. Cos değerinin sıfır çıkarması iki doküman arasında herhangi bir ilişkinin olmadığı, bir çıkarması iki dokümanın birbirinin aynısı olduğu anlamına gelmektedir. Kosinüs benzerliği formülü aşağıda verilmektedir.

$$\cos \phi = \frac{d1.d2}{||d1||.||d2||} \quad (3.2)$$

$$\Theta = \arccos ((A \cdot B) / (||A|| ||B||)) \quad (3.3)$$

#### 3.3.4 Gereksiz Verilerin Temizlenmesi

Çok sık tekrar eden zamirler, edatlar ve bağlaçlar metinlerin birbirleriyle karşılaştırılmasında ayrıştırıcı özelliğe sahip olmadıklarından bunların belirlenip temizlenmesi gerekmektedir (Salton G. ve ark., 1975). Bu aşamada doküman içerisinde yarar sağlamayacak terimler belirlenir. Böylece dokümanda bulunması istenmeyen gereksiz veriler temizlenmiş olur. Bu kelimeler İngilizcede “or”, “and”, “am/is/are” gibi tek başına bir anlama sahip olmayan terimler olabilirken, Türkçe için de “ve”, “bir”, “ben”, “veya”, “o”, “ama” gibi kelimeler olabilmektedir (Kılınç D. ve ark., 2015).

#### 3.3.5 Kelime Köklerinin Tespiti

Kelimeler eklerle birleştiğinde içerik olarak farklı bir yapıda olsalar da aynı anlama sahiptirler. Anlam olarak doğru bir ifadeye ulaşabilmek için ekler kelimelerden çıkartılarak kök hali ile ele alınmaktadır. Örnek olarak İngilizcede;

Pens: pen: s eki, Walking: walk: ing eki atılarak kelime köküne ulaşılmaktadır.

İngilizce kök bulmak için genellikle ‘Porter’s Stemming Algorithm’ kullanılmaktadır (Porter M. F., 1980). Türkçe için bu konuda Oflazer’in çalışmaları mevcuttur (Oflazer K., 1994).

Kelime köklerinin bulunması bir anlamda özniteliklerin belirlenmesi anlamına gelmektedir. Bu öznitelikler tüm veri kümesini ifade eden gerçek değerlerdir. Oluşturulacak bir matris üzerinde öznitelikler sütun matrisi olarak işlem görmektedirler (Şirin Y. ve Kutlugün M.A., 2017). Özniteliklerin benzersiz olması sistemin başarımı açısından oldukça önemlidir. Öznitelikler elde edildikten sonra terim ağırlıklandırma işlemi uygulanmaktadır.

### 3.3.6 Özellik Belirleme

Metin sınıflandırmada ilk olarak metnin özellik vektörleri çıkarılır. Her bir dokümanın kendine has bazı özelliklerinin belirlenmesi gerekir. Özellik çıkarımında yaygın olarak N-gram ve Terim Frekans istatistikleri kullanılmaktadır (Pehlivan, 2014).

### 3.3.7 N-Gram Modeli

N-gram, bir karakter (harf) kümesinin ‘n’ adet karakteri ile oluşan dilimidir. N-gram kullanan sınıflandırma yöntemi, seçilen ‘N’ değerine göre ‘2-gram’, ‘3-gram’, ‘4-gram’ vb. kullanılabilir. Karakter n-gram veya kelime n-gram olarak iki türü vardır (Pehlivan, 2014).

Karakter n-gram için “*Özellik Belirleme*” ifadesi:

**2-gram olarak:** ‘Öz’, ‘ze’, ‘el’, ‘ll’, ‘li’, ‘ik’, ‘k\_’, ‘\_B’, ‘Be’, ‘el’, ‘li’, ‘ir’, ‘rl’, ‘le’, ‘em’, ‘me’,

**3-gram olarak:** ‘Öze’, ‘zel’, ‘ell’, ‘lli’, ‘lik’, ‘ik\_’, ‘\_Be’, ‘Bel’, ‘eli’, ‘lir’, ‘irl’, ‘rle’, ‘lem’, ‘eme’ belirtilir.

Kelime n-gram için “*Makine Öğrenme Yoluyla Ses Sentezleme*” ifadesi:

**Kelime 2-gram olarak:** ‘Makineöğrenme’, ‘öğrenmeyoluyla’, ‘yoluylases’, ‘sessentezleme’

**Kelime 3-gram olarak:** ‘Makineöğrenmeyoluyla’, ‘öğrenmeyoluylases’, ‘yoluylasesentezleme’ şeklinde kullanılmaktadır.

Her n-gram’ın metinlerde ve tüm veri kümesinde kaç defa geçtiğinin istatistiki bilgisi tutulmaktadır. Doküman kümesinde bulunan n-gramlar en yüksekten en düşük frekansa doğru sıralanarak özellik vektörleri elde edilmektedir.

### 3.3.8 Terim Frekansları

Bu yöntemde metinler içindeki terimler frekansları alınarak ifade edilmektedir. Bu terimler direk kelimelerin kendisi, kelime kökleri veya karakter gramları şeklinde belirlenebilmektedir. Bu şekilde ele alınarak oluşturulan bir matriste satırlarda metinler, sütunlarda ise terimler yer almaktadır. Aşağıda gösterimi verilen örnek matriste  $D_1, D_2, \dots, D_n$  : dokümanları,  $T_1, T_2, \dots, T_n$  : doküman içindeki terimleri ifade etmektedir.

$$\begin{pmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & d_{11} & d_{12} & \dots & d_{1t} \\ D_2 & d_{21} & d_{22} & \dots & d_{2t} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & d_{n1} & d_{n2} & \dots & d_{nt} \end{pmatrix}$$

Şekil 3.5. Örnek bir doküman-terim frekans matrisi

### 3.3.9 Terim Ağırlığı

Terim ağırlıklandırma (TA) ile her bir terim; ilgili terimin önemini ölçen ve gözlemlendiği dokümanın sınıflandırılmasına yaptığı katkıyı belirten bir ağırlık değeri ile ilişkilendirilir (Patra, A. ve Singh, D., 2013). Denklem olarak gösterimi aşağıdaki gibidir.

$$d = (w_1, w_2, \dots, w_{|T|}) \quad (3.4)$$

w: ncı terimin ağırlık değeri, T: bir doküman içindeki benzersiz terimlerin sayısı

TA sınıflandırmanın başarımına direk etki etmektedir. Bir terim ağırlıklandırılırken terim frekansı (TF: Term Frequency), ters doküman frekansı (IDF: Inverse Document Frequency) ve normalizasyon olmak üzere üç etkenden yararlanılmaktadır

(Salton, G. ve Buckley, C. , 1988). Bu bileşenlerden TF bir t teriminin bir d dokümanındaki ağırlığını, IDF ise veri kümesi içindeki ağırlığını temsil etmektedir.

$$TF(t) = \frac{t \text{ teriminin bir dokümanda geçme sayısı}}{\text{Dokümandaki toplam terim sayısı}} \quad (3.5)$$

$$IDF(t) = \log\left(\frac{\text{Toplam doküman sayısı}}{\text{İçerisinde t terimini bulunduran toplam doküman sayısı}}\right) \quad (3.6)$$

$$w = tft * idft \quad (3.7)$$

Bu ağırlıklandırma metotları dışında sıklıkla kullanılan ve TF ile IDF birleşimi olarak ortaya çıkan terim frekansı-ters doküman frekansı (TF-IDF) ve türevi metotlar da kullanılmaktadır.

### 3.3.10 Boyut Azaltma

Makine öğrenmesinde boyut azaltma çok boyutlu uzayı daha az miktardaki bir uzay ile ifade etme işlemidir. Yani mevcut özniteliklerden veriyi tanımlayan bir öznitelik alt kümesinin seçilmesidir. Doküman kümesinden ayırt edici özelliği bulunmayan öznitelikler tespit edilerek çıkarılmak suretiyle boyut azaltılmış olur (Şirin Y. ve Kutlugün M.A., 2017). Boyut azaltma başarımı etkileyen önemli bir adımdır. Boyut azaltma sayesinde, ideal işlem sürelerinde başarımı artıran sonuçlar elde edebilecek alt kümeler meydana getirildiğinden, sürecin etkinliğini ve doğruluğunu artırmak amacıyla daha kesin sonuçlara ulaşılabilmektedir. Bu konuda öznitelik seçme ve öznitelik çıkarma şeklinde iki yaklaşım vardır.

Öznitelik seçme yaklaşımı ile yüksek boyutlu uzay sorununu çözmek için sistem eğitim aşamasından önce mevcut özniteliklerden veriyi tanımlayan bir öznitelik alt kümesi seçilmektedir (Dasgupta, A. ve ark., 2007). Bu işlem veriyi görselleştirme ve anlamayı daha kolay hale getirme, ölçüm ve veri depolama gereksinimlerinde azalma sağlama, eğitim ve test zamanlarını kısaltma gibi faydalar sağlayarak performans artışına etki eder (Guyon, I. ve Elisseeff, A., 2003).

Öznitelik çıkarma yaklaşımında ise öznitelik boyutu daha etkili ve daha küçük boyutlu bir alt uzaydan oluşur (Li, Y. H. ve Jain, A. K., 1998). Bu yaklaşımlar sayesinde veri kümesinden ilgisiz (bilgi içermeyen) ve gürültülü olarak adlandırılan öznitelikler elenir ve öznitelik uzay boyutu düşürülür. Bu durum sınıflandırmada



eđitim ve test ařamalarında performans ve zaman kazancı bakımından olumlu etki oluřturmaktadır (Çoban, 2016).

Literatürde kullanılan birçok öznitelik seçme metodu olmakla beraber Ki-Kare (CS: Chi-Square), Karřılıklı Bilgi (MI: Mutual Information), Bilgi Kazanımı (IG: Information Gain) ve Doküman Frekansısı (DF: Document Frequency) sıklıkla tercih edilen metotlardandır (Yang, Y. ve Pedersen, J. O., 1997), (Forman, 2003).

### 3.3.11 Doküman Frekansısı (Document Frequency)

Doküman frekansısı, dokümanda bir terimin kaç defa görüldüğünü ifade eder. Bu metotta belirlenen eşik deđerinin altında kalan terimler atılır. Bu yöntem ile daha az kullanılan terimlerin bulunması gerçekleştirilebilir. Yang ve Pedersen bu yöntemin metin sınıflandırma üzerinde inceleyip, farklı yöntemlerle karşılaştırıldığında hesaplama karmaşıklığı bakımından en sade ve en etkili yöntem olduğunu belirlemişlerdir (Uzun, 2007).

### 3.3.12 Performans Ölçümü

Performans deđerlendirme işlemi için genel olarak karmaşıklık matrisinden (confusion matrix) yararlanılmaktadır. Bu matrise göre Doğru Pozitif (DP), Doğru Negatif (DN), Yanlış Pozitif (YP) ve Yanlış Negatif (YN) olarak adlandırılan deđerler sonucunda "Pozitif" ve "Negatif" olarak iki sınıfa ait dört farklı deđer elde edilmektedir. Bu deđerler, verilen bir veri kümesinde gerçekte ait olduğu sınıfı ve tahmin sonucu öngörülen sınıf ile eşleşen ve eşleşmeyen örneklerin sayısını göstermektedir. Bir veri kümesinde gerçekte pozitif sınıfa ait olup, sınıflandırma neticesinde yine pozitif sınıfta yer alan bir örneğe doğru pozitif (True Positive, TP) adı verilir. Gerçekte negatif sınıfa ait olup sınıflandırma sonucunda negatif sınıfta yer alan doğru negatif (True Negative, TN) adını alır. Gerçekte negatif sınıfta olmasına rağmen pozitif sınıfta tahmin edilen örneğe yanlış pozitif (False Positive, FP) denir. Pozitif sınıfta olmasına rağmen negatif sınıfta tahmin edilen örnekler ise yanlış negatif (False Negative, FN) olarak isimlendirilmektedir (Biricik, 2011). Ařağıdaki Tablo 3.1'de tüm durumlar gösterilmektedir.

Tablo 3.1. Karmaşıklık matrisi

		Gerçek Sınıf	
		Doğru Sınıf	Yanlış Sınıf
Tahmin Edilen Sınıf	Doğru Sınıf	DP	DN
	Yanlış Sınıf	YP	YN

Doküman sınıflandırmada sıklıkla kullanılan karşılaştırma yöntemleri kesinlik ve anma ölçüleri ile belirlenmektedir. Kesinlik değeri bir sınıftaki doğru olarak sınıflandırılan dokümanların sayısının, o sınıftaki toplam doküman sayısına oranını; anma değeri ise bir sınıftaki doğru olarak sınıflandırılan dokümanların sayısının, sistemin o sınıf olarak tespit ettiği toplam doküman sayısına oranını verir. Kesinlik ve anma değerleri tek başlarına anlamlı bir karşılaştırma sonucu için yeterli değildir. Her ikisini beraber değerlendirmek daha doğru sonuçlar vermektedir. Kesinlik ve anma değerlerinin her ikisinin de aynı anda iyi olması amacıyla F-ölçüsü değeri hesaplanmaktadır (van Rijsbergen C. J., 1979), (Manning, D. C. ve Schutze H., 1999). Tüm bu değerlere ait formüller aşağıda sunulmaktadır.

$$Kesinlik = \frac{\text{Doğru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif sınıflandırılmış örneklerin sayısı}} = \frac{DP}{DP+YP} \quad (3.8)$$

$$Anma = \frac{\text{Doğru sınıflandırılmış pozitif örnek sayısı}}{\text{Pozitif örnek sayısı}} = \frac{DP}{DP+YN} \quad (3.9)$$

$$F - \text{Ölçüsü} = \frac{2 \times Kesinlik \times Anma}{Kesinlik + Anma} \quad (3.10)$$

### 3.4 Bazı Önemli Makine Öğrenmesi Algoritmaları

Makine öğrenmesi ile ilgili birçok algoritma ve yöntem geliştirilmiştir. Bunlar çok çeşitli alanlarda kullanılmaktadırlar. Bazı önemli makine öğrenmesi algoritmaları hakkında aşağıda detaylı bilgiler sunulmaktadır.

#### 3.4.1 K-Ortalama (Means) Algoritması

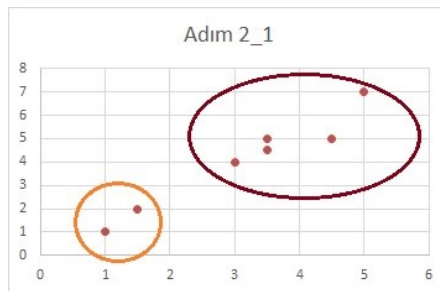
K-ortalama algoritması, veri kümesindeki n adet nesnenin k tane kümeye bölünmesini sağlamaktadır. Kümeleme sonucunda kümelerin içinde yer alan nesnelere arasındaki benzerliklerin çok olması, diğer kümelerin nesnelere ile

arasındaki (inter-cluster) benzerliklerin çok düşük olması beklenmektedir (Alzand, H.R.A. ve Karacan H., 2014).

K-ortalama algoritmasının iki temel bileşeni vardır. Birinci bileşen için k merkez değerleri rastgele biçimde belirlenmeli yani, k merkez değeri girdi olarak sunulmuş olmalıdır. Diğer bileşen için ise veri kümesindeki her nesne kendisine en yakın merkez noktanın olduğu demete dâhil olmalıdır (Chen, J. ve ark., 2009).



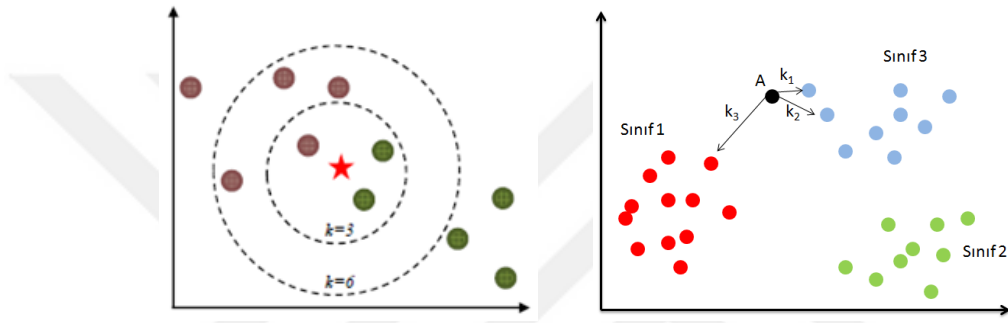
K-ortalama algoritmasında büyük hacimdeki veri kümelerinin en az olduğu durumda ilgili merkezin kümesine dâhil edilmiş olmaktadır. Daha sonra, her bir kümenin ortalama değeri ayrı ayrı hesaplanmakta ve yeni küme merkezleri belirlenmektedir. Aynı şekilde tekrardan nesne-merkez uzaklıkları hesaplanarak ortaya çıkan her nesne için tüm k merkeze uzaklık değeri göze alınarak hangi kümeye ait olacağı belirlenmektedir. Nesnenin hangi merkeze uzaklığı en düşük ise o merkezin kümesine dâhil olacak şekilde işlemler sürdürülmektedir (Jiawei, 2006).



### 3.4.2 K - En Yakın Komşu Algoritması

K-En Yakın Komşu algoritmasında (k-NN: k-Nearest Neighbor) bir nesnenin sınıfını belirlemek için önceden sınıfları bilinen ve eğitim kümesi olarak adlandırılan

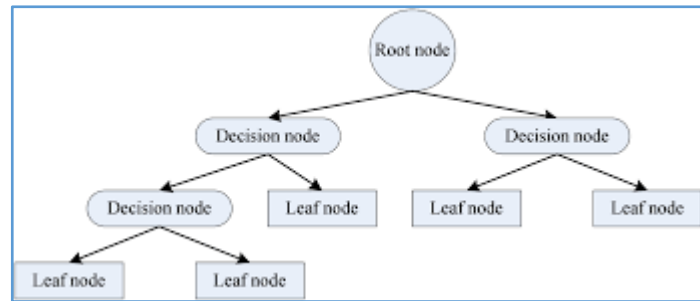
kümedeki  $k$  adet elemandan yararlanılmaktadır. Sınıfları bilinmeyen bir nesne her bir sınıftaki diğer tüm nesnelerle karşılaştırılmaktadır. Bu nesneye en yakın  $k$  tanesi seçilmektedir. Seçilen nesneler adet olarak en çok hangi sınıfa ait ise sınıfı belirlenmek istenen nesne de o sınıfa dâhil edilmektedir. Nesneler arasındaki uzaklığın hesaplanması için genellikle Öklid uzaklık formülü veya Kosinüs benzerliği formülü kullanılmaktadır (Özkan, 2008).  $K$ -en yakın komşu algoritması oluşturulurken, ilk önce  $k$  değerine karar verilmektedir. Bu  $k$  değeri sınıfı henüz bilinmeyen yeni örneğin en yakın  $k$  tane komşusuna bakılarak sınıfının belirlenmeye çalışıldığını ifade eder (Cover, T.M. ve Hart, P.E., 1967 ).



Şekil 3.8. K-En yakın komşu sınıflandırması modeli

### 3.4.3 Karar Ağaçları (Decision Trees: DTs)

Karar ağaçları, verileri ağaç yapısı şeklinde gösteren bir sınıflandırıcıdır. Bir karar ağacı düğüm, dal ve yapraklardan oluşmaktadır. Düğüm soruları temsil ederken, dal bu sorulara verilebilecek cevapları, yaprak ise kararlaştırılan sınıfı temsil etmektedir (Yılmaz R. , 2013).



Şekil 3.9. Örnek karar ağacı modeli

Karar ağacı, sınıflandırma yapmak için kök düğümünden (root node) başlayarak yaprak düğümlere (leaf node) doğru hareket etmektedir. Karar ağacı ile sınıflandırma, bilgi

öğrenmek için tümevarımsal bir yaklaşımdır. Örneklerinin özelliklerine göre arama yapmakta ve verilen örnekleri en iyi biçimde ayırarak yeni özellikleri elde etmektedir. Özellik eğitim kümesini en iyi şekilde sınıfladığı zaman durmaktadır. Sınıflandırma uygun bir biçimde yapılamadığında, en iyi özelliği bulmak için bir özelliğe ait muhtemel değerlerin sayısı kadar bölümlere ayrılmış alt kümeler üzerinde tekrarlı şekilde işlem yapmaya devam etmektedir. Bu algoritma, en iyi değerleri seçerken önceki seçimleri yeniden ele almamaktadır. Bundan dolayı hatalı sınıflandırma durumları oluşabilmektedir (Uzun, 2007). Karar ağacının ana işlevi, ağaçtaki her düğümü test ederek özellik belirlemektir.

#### 3.4.4 Naive Bayes Sınıflandırıcı

Naive Bayes, metin sınıflandırmada çok sık kullanılan ve diğer bazı sınıflandırıcılara göre daha doğru sonuçlar veren bir sınıflandırma yöntemidir. Dokümanın sınıfı kelimelerin ve sınıfların birleşik olasılıkları kullanılmak suretiyle belirlenmektedir (Pehlivan, 2014). Naive Bayes yöntemi, sınıf belirlemede sözcüklerin ve sınıfların koşullu olasılıklarını kullanmaktadır. Bir sınıflandırıcı Bayes ağı kendisine verilen sorulara karşılık sonuçlar üretmektedir. Bu ağın kullanılabilmesi için olasılık modeline ihtiyaç duyulmaktadır (Yılmaz R. , 2013). Bayes teoreminin genel ifadesi aşağıdaki gibidir.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \quad (3.11)$$

**P(A)** ve **P(B)** : A ve B olaylarının olasılıkları,

**P(A|B)** : B olayının gerçekleşmesi durumunda A olayının meydana gelme olasılığı,

**P(B|A)** : A olayının gerçekleşmesi durumunda B olayının meydana gelme olasılığıdır.

Naive Bayes ağları ile otomatik doküman sınıflandırma yapmak için önce dokümanlar ve sınıflar belirlenmektedir. Elde edilen verilerden kelimelerin dokümanlardaki olasılıkları hesaplanarak dokümanın sınıfı tahmin edilmektedir.  $D=\{d_1, d_2, \dots, d_n\}$  Sınıfı bilinmeyen dokümanlar,  $C_1, C_2, \dots, C_m$  de sınıflar olarak temsil edildiğinde, sınıfı belirlenecek olan dokümanın olasılık denklemi aşağıdaki gibi olmaktadır (Yılmaz R. , 2013).

$$P(C_i|D) = \frac{P(D|C_i) \times P(C_i)}{P(D)} \quad (3.12)$$

**Avantajları:** Eğitim ve değerlendirme işlemleri oldukça hızlıdır ve gerçek hayattaki problemlerde iyi derecede sonuçlar vermektedir (Uzun, 2007).

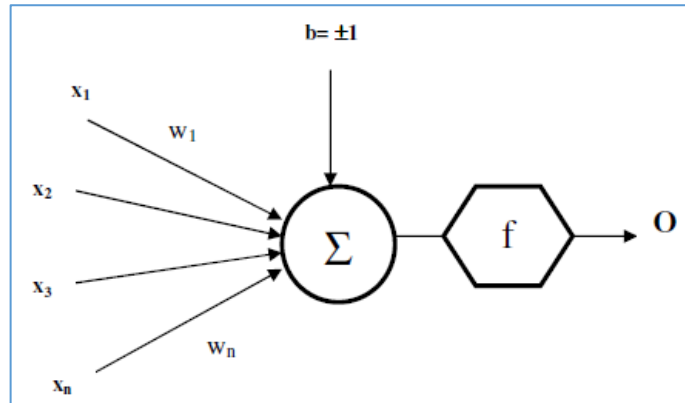
**Dezavantajları:** Çok karmaşık olan sınıflama problemlerinin çözümünde yetersiz kalabilmektedir (Uzun, 2007).

### 3.4.5 Yapay Sinir Ağları

Yapay sinir ağları (YSA) ile ilgili ilk çalışmalar 1950 yıllarında Frank Rosenblatt tarafından basit nöron yapısına dayalı bir hesaplama modeli önermesi ile başlamıştır (Rosenblatt F. , 1959). Daha sonra perceptron adı verilen tek katmanlı ilk YSA modeli geliştirilmiştir (Rosenblatt F. , 1962).

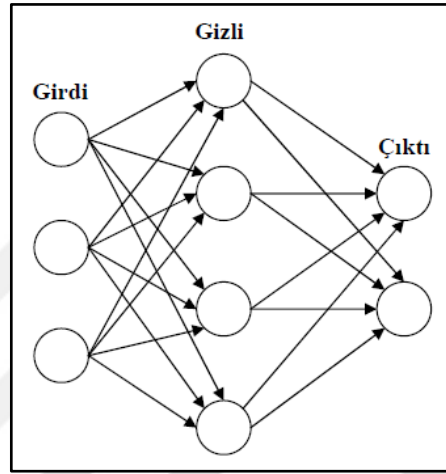
YSA, insan beyninin sinir sistemini model alan bir çalışma prensibine sahip yöntemdir. İnsan beyninin öğrenmeyle yeni bilgiler oluşturabilme, keşif yapabilme, var olan bilgiler ile olaylar hakkında yorumlama yapabilme, karar verebilme, olaylar arasında ilişki kurabilme gibi özellikleri gerçekleştirebilmek için tasarlanmıştır (Tekeli K. ve Aşlıyan R., 2016).

YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde birleştirilmesinden oluşmakta ve genellikle katmanlar halinde düzenlenmektedirler. Donanım olarak elektronik devrelerle tasarlanabilmektedirler. YSA, bir öğrenme süreci sonrası bilgiyi toplama ve hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama yeteneğine sahiptir. Ayrıca, öğrenme süreci ile arzu edilen amaca ulaşabilmek için YSA ağırlıklarının güncellenmesini sağlayan öğrenme algoritmalarını içermektedir (Haykin, 1994).



Şekil 3.10. Temel YSA hücresi (Taşova, 2011)

Bir YSA belirli bir amaç için oluşturulmakta ve insanlar gibi örnekler sayesinde öğrenmektedir. Öğrenme, Yapay Sinir Ağlarında tekrarlanan girdiler ile ağı kendi yapısını ve ağırlıklarını güncellemesi sayesinde gerçekleşmektedir (Wu, C.H. ve Tsai, C.H., 2009). Yapay sinir ağlarının Tek Katmanlı (Perceptron) ve Çok Katmanlı olmak üzere iki türü vardır. Çok Katmanlı Algılayıcı Modeli, bir girdi katmanı, bir veya daha fazla gizli katman ve bir de çıktı katmanından oluşmaktadır. Aşağıda Çok Katmanlı Algılayıcı Modelinin genel yapısı verilmiştir (Tekeli K. ve Aşlıyan R., 2016).



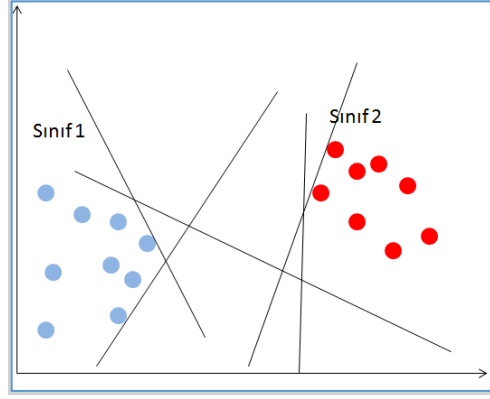
Şekil 3.11. Çok katmanlı YSA modeli

Girdi Katmanı, dışarıdan alınan bilgileri ara katmanlara transfer etmektedir. Gizli Katman, girdi katmanından gelen bilgileri işleyerek çıktı katmanına göndermekte ve bir ağ içinde birden fazla sayıda gizli katman olabilmektedir. Çıktı Katmanı, gizli katmandan gelen bilgileri işleyerek girdi katmanında verilen girdi kümesi için üretmesi gereken çıktıyı vermektedir. Son olarak elde edilen çıktıları dış dünyaya göndermektedir (Telcioğlu, 2007).

#### 3.4.6 Destek Vektör Makinesi (Support Vector Machine: SVM)

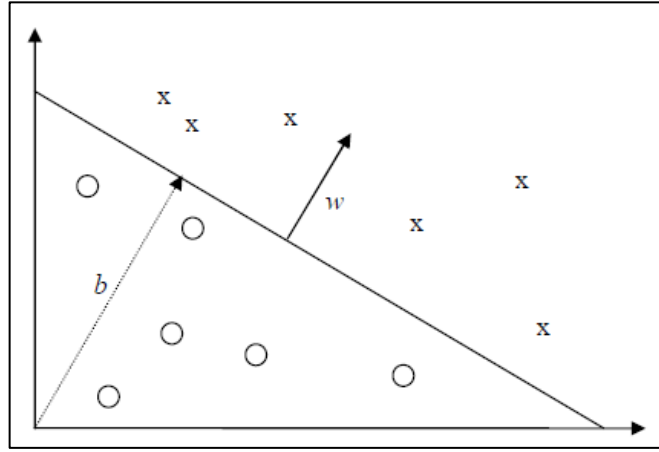
Optimizasyona dayalı bir sınıflandırma modeli olan destek vektör makineleri uygun bir karar fonksiyonu ile sınıfları birbirinden ayırt edebilecek bir düzlem belirtmektedir. Destek vektör makinelerinin amacı bu düzlemlerden iki sınıfı en iyi ayıran hiper düzlemi belirleyebilmektir. Bunun için öncelikle veri kümesinin doğrusal olarak ayrılabilir veya ayrılmaz olduğuna karar verilerek buna uygun sınıflandırma işlemi gerçekleştirilmektedir (Yılmaz R. , 2013).

İki farklı veri kümesini lineer olarak birbirinden ayıran bir örnek sınıflandırma aşağıda verilmiştir.



Şekil 3.12. Destek vektör makinesi sınıflandırma modeli

Destek vektör makineleri büyük boyutlu verileri sınıflandırmada oldukça yaygın kullanılan bir yöntemdir. Düşük boyutlu ve lineer olarak ayıramayacak olan bir veri kümesini, daha yüksek bir boyuta taşıyarak bir düzlem yardımıyla ayırma işlemi sağlamaktadır. Destek vektör makinelerinin çalışma mantığını anlamak için, lineer ayırmanın nasıl yapıldığı incelenmelidir (Kecman, 2001), (Herbrich, 2002).



Şekil 3.13. Lineer destek vektör makinesi sınıflandırma modeli

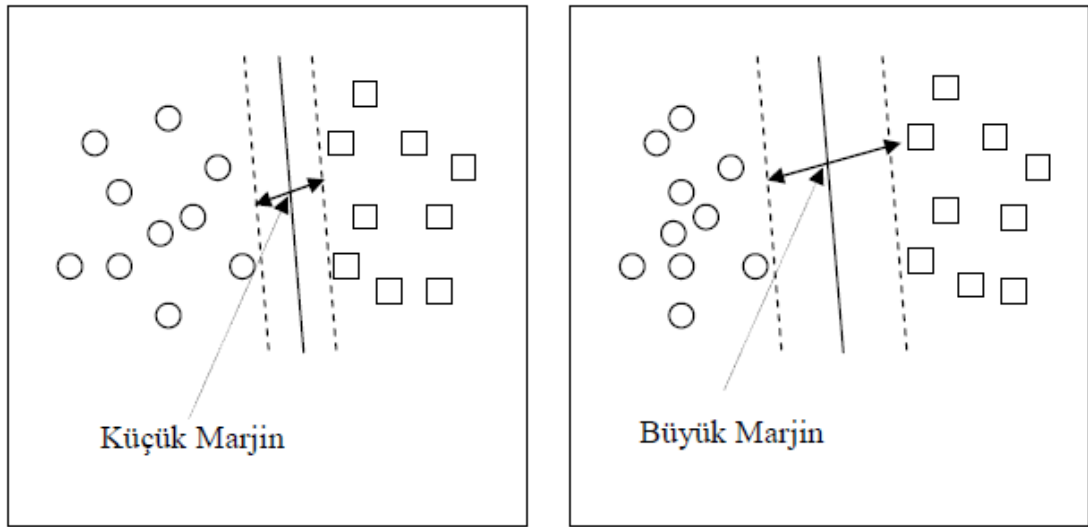
Yukarıdaki şekilde gösterilen Fisher Lineer Diskriminantı (Fisher Linear Discriminant) ile iki farklı veri grubu birbirinden lineer olarak ayrılmaktadır. Bu lineer doğru veya ayırım fonksiyonu için aşağıdaki formül kullanılmaktadır (Uzun, 2007).



$$f(x) = \sum_{n=1}^n w_i x_i + b \quad (3.13)$$

Formülde x değeri ayırım yapılacak girdi verisini, b değeri eşik değer veya orjine uzaklığı ifade etmektedir. w değeri ise ağırlık matrisi veya veriye bağlı bir vektörü tanımlamaktadır. Bu vektör veri eğitim aşamasında şekillenmektedir. Eğer  $f(x) \geq 0$  ise x vektörünün sınıfı pozitif, tersi ise negatif olarak değerlendirilmektedir (Uzun, 2007).

Çeşitli destek vektör makineleri mevcuttur ve verinin gürültü içerip içermemesine bağlı olarak kullanılan yaygın iki yöntem vardır. Bunlar maksimal marjin (maximal margin) ve esnek marjin (soft margin) yöntemleridir (Uzun, 2007).



Şekil 3.14. Maksimal marjin ve esnek marjin (Uzun, 2007)

Bu yaklaşımlarda en iyi sınıflandırmayı gerçekleştirebilmek için yapılması istenen farklı sınıftan örnekler arasındaki uzaklığı en büyüyen hiper düzlemin belirlenmesidir.

## 4 DENEYSEL ÇALIŞMA

### 4.1 Bir Veri Kümesi Üzerinde Bazı Makine Öğrenmesi Algoritmalarının Karşılaştırılması

Bu deneysel çalışma, bir makine öğrenmesi uygulaması olan “WEKA” programı kullanılarak gerçekleştirilmiştir. Seçilen bir veri kümesi program içerisinde varsayılan olarak tanımlı makine öğrenmesi algoritmalarına ayrı ayrı uygulanmıştır. Bu algoritmalar arasındaki farklılıklar ve başarımları gözlemlenmiştir. Uygulamanın amacı, aynı veri kümesinde farklı makine öğrenmesi algoritmalarının başarımları açısından karşılaştırılabilmesi, görsel olarak sonuçların izlenebilmesi ve hızlıca değerlendirilebilmesine olanak sağlamasıdır. Seçilen veri kümesinden elde edilen sonuçların başka veri kümelerinde aynı algoritmalar ile daha farklı sonuçlar verebileceği unutulmamalıdır. Kısaca, bu veri seti için başarımları yüksek olan bir algoritma diğer veri setlerinde düşük sonuçlar verebilir.

#### 4.1.1 WEKA

WEKA, makine öğrenimi amacını yerine getirmek için 1993 yılında, “University Of Waikato” tarafından geliştirilmiştir. Günümüzde sık kullanılan birçok makine öğrenimi algoritmalarını içermektedir (Pehlivan, 2014). WEKA içerdiği birçok makine öğrenmesi yöntemini çeşitli şekillerde test edilebilen, Java platformunda geliştirilmiş açık kaynak kodlu bir veri madenciliği uygulama programıdır (Biricik, 2011). Java dilinde geliştirilmiş olması ve kütüphanelerinin Java dilinde yazılan başka projelere entegre edilebilmesi olanakları sunması sayesinde kullanımı daha da yaygınlaşmıştır. WEKA, tamamen modüler bir tasarıma sahiptir ve içerdiği özelliklerle veri kümeleri üzerinde görselleştirme, veri analizi, iş zekâsı uygulamaları, veri madenciliği gibi işlemlere de olanak sağlamaktadır. Sınıflandırma, kümeleme, ilişkilendirme işlemleri haricinde veri ön işleme için çeşitli filtreleme ve boyut indirgeme yöntemlerine de sahiptir. Ayrıca işlem sonuçlarının görsel olarak görüntülenmesine olanak tanır. WEKA program kütüphanesinde çeşitli veri

kümelerini içeren pek çok dosya üzerinde çalışan hazır fonksiyonlar da bulunmaktadır. WEKA ile aşağıdaki işlemler kolaylıkla gerçekleştirilebilmektedir.

1. Sınıflandırma (*Classification*)
2. Kümeleme (*Clustering*)
3. İlişkilendirme (*Association*)

#### 4.1.2 WEKA Dosya Yapısı

WEKA yazılımı, kendisine özel bir ‘.arff’ dosya uzantısı ile kullanılmaktadır. ARFF ismi, İngilizce olan ‘‘*Attribute Relationship File Format*’’ kelimelerinin baş harflerinden oluşmuştur. ARFF dosya biçimi metin yapısında kullanılan bir biçimdir. Dosyanın ilk satırında dosyadaki ilişki (*relation*) yer alırken ikinci satırdan itibaren veri kümesindeki özellikler (*attributes*) yer almaktadır. Özelliklerin sonrasında veri kümesi başlamakta ve veri kümesindeki her satır bir örneğe (*instance*) karşılık gelmektedir. Ayrıca veri kümesindeki her örnek virgül ile birbirlerinden ayrılmaktadır. Tablo 4.1’de örnek bir .arff dosyası yapısı verilmiştir.

Tablo 4.1. Örnek .arff dosya yapısı

```
@relation Textdata100

@attribute flow real
@attribute librari real
@attribute boundari real
@attribute inform real
@attribute pressur real
@attribute drag real
@attribute air real

@attribute class {CISI,CRAN,MED}

@data
0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,CISI
0,2,0,0,0,2,2,0,0,0,0,1,0,0,0,0,0,1,0,0,0,2,0,0,0,0,CISI
0,0,0,0,1,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0,1,0,1,0,0,0,CRAN
0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,CRAN
2,0,8,0,0,0,1,0,1,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MED
0,0,8,0,0,1,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MED
```

#### 4.1.3 Deneysel Çalışma

Bu çalışmada literatürde “classic3” veri seti olarak adlandırılan, WEKA programında kullanılan “.arff” dosya uzantısına sahip akışkanlar dinamiği, bilimsel endeksler ve tıbbi konuları içeren, 3 farklı türe ait 101 adet özniteliğe sahip, 3830 tane örnekten meydana gelen veri setinden yararlanılmıştır. Bu veri setine aşağıdaki algoritmalar uygulanmıştır;

- K - En yakın komşu algoritması
- Karar Ağaçları algoritması
- Naive Bayes sınıflandırıcı
- Destek Vektör Makineleri

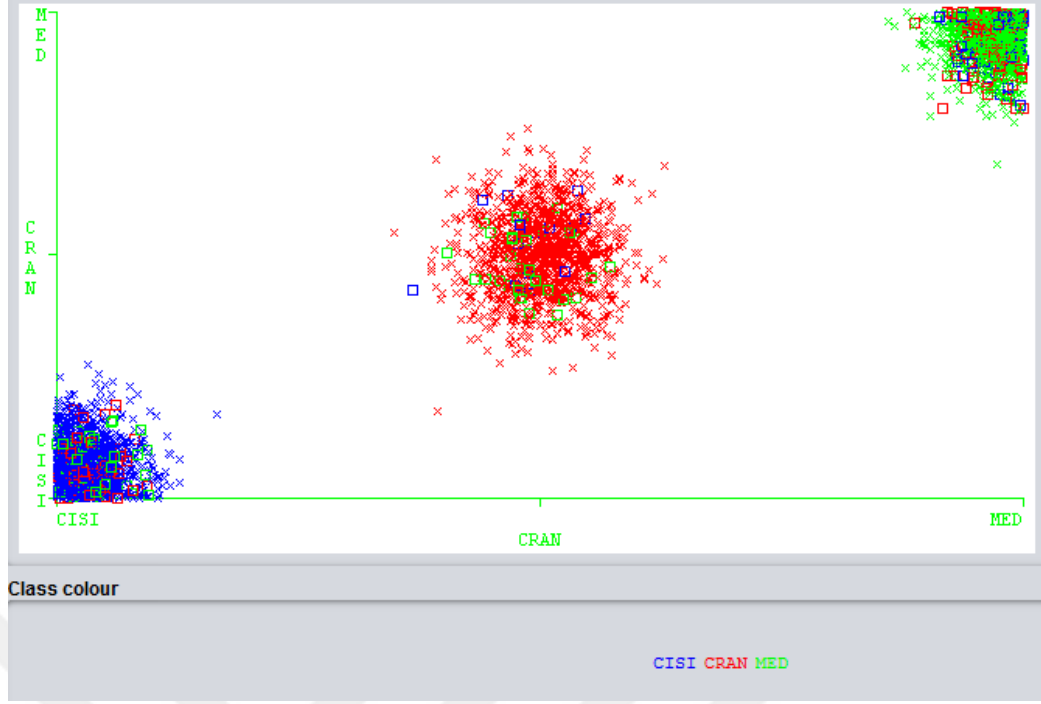
#### 4.1.4 K - En Yakın Komşu Algoritması

Bu algoritma  $k=1$  değeri için uygulanmıştır. Bu değere göre algoritma, örnekler içinden kendisine en benzer nesneye (tek nesneye) bakmak suretiyle değerlendirme yapmakta ve sınıfını belirlemek istediği nesneyi baktığı bu nesnenin sınıfına göre atamaktadır.

Tablo 4.2. K-En yakın komşu algoritmasının başarımı

CISI	CRAN	MED	
1348	17	67	CISI
79	1162	154	CRAN
64	34	905	MED
Doğru Sınıflandırılan:			3415
Yanlış Sınıflandırılan:			415
Başarım Oranı (%) :			89,164

K-En yakın komşu algoritmasına göre yapılan sınıflandırma neticesinde verilerin farklı sınıflara dağılım durumları Şekil 4.1’de görülmektedir.



Şekil 4.1. K-En yakın komşu algoritmasının veri dağılımı

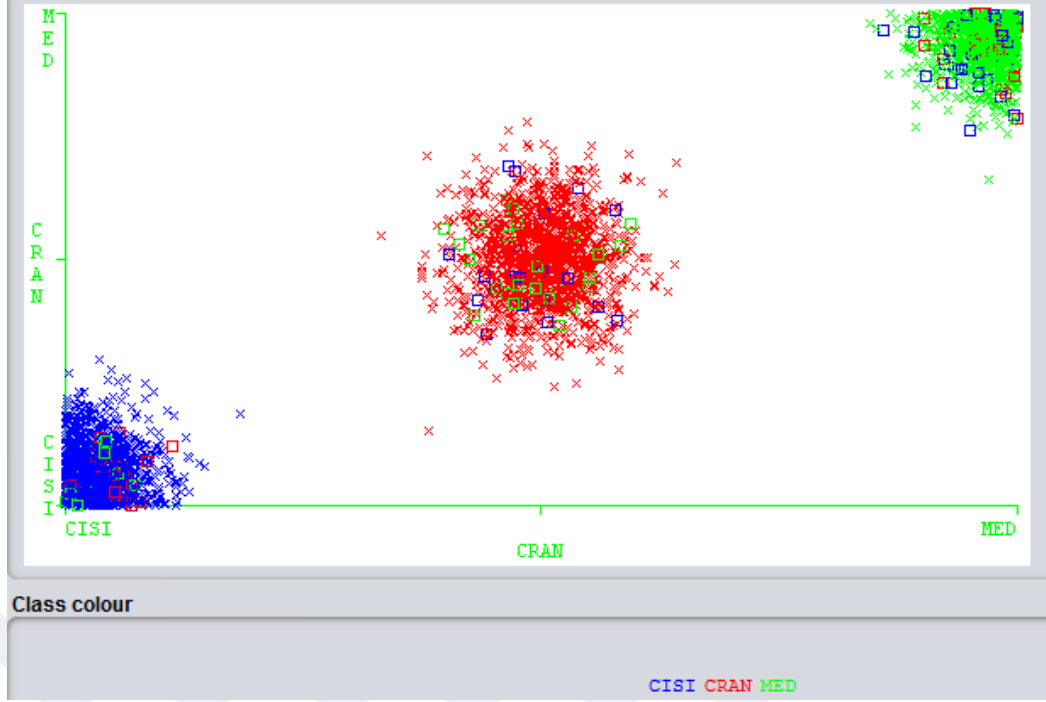
Bu algoritma, farklı k değerlerine göre de sınıflandırılabilir ve başarımları en iyi sonucu veren k değeri kullanılabilir.

#### 4.1.5 Karar Ağaçları Algoritması

Tablo 4.3. Karar ağaçları algoritmasının başarımları

CISI	CRAN	MED	
1367	21	44	CISI
43	1315	37	CRAN
19	28	956	MED
Doğru Sınıflandırılan:			3638
Yanlış Sınıflandırılan:			192
Başarımları Oranı (%) :			94,987

Karar ağaçları ile yapılan sınıflandırma neticesinde verilerin farklı sınıflara göre dağılımları Şekil 4.2’de gösterilmiştir.



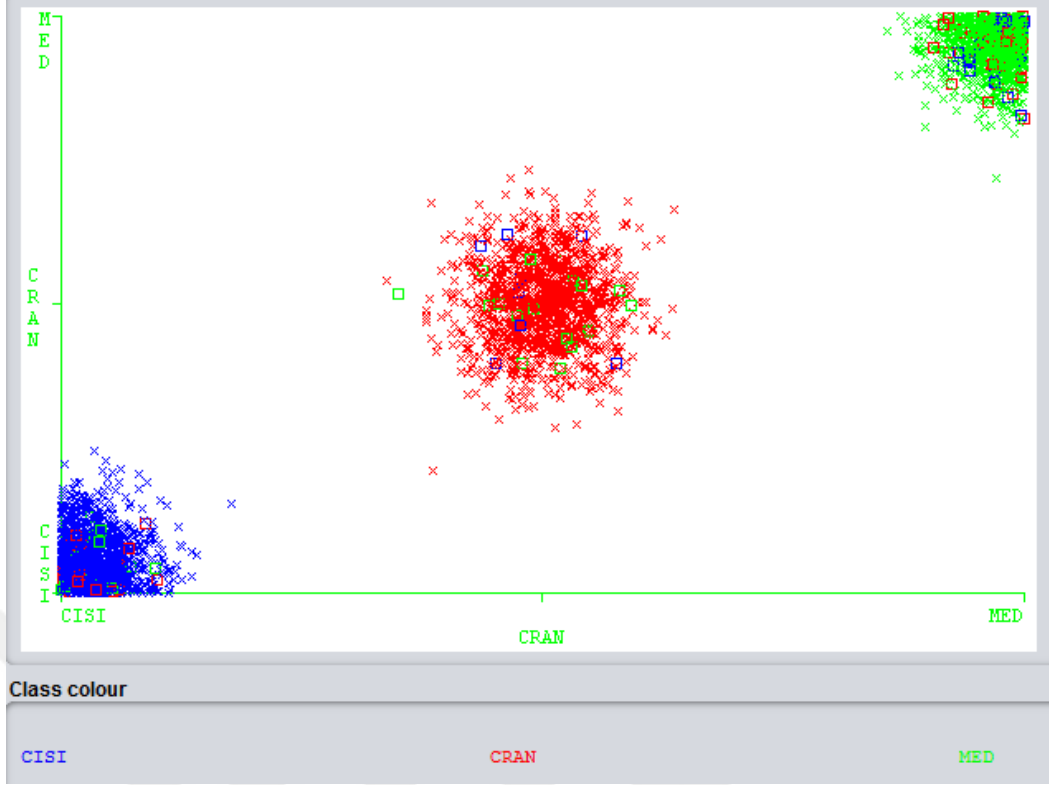
Şekil 4.2. Karar ağaçları algoritmasının veri dağılımı

#### 4.1.6 Naive Bayes Sınıflandırıcı

Tablo 4.4. Naive bayes algoritmasının başarımı

CISI	CRAN	MED	
1391	9	32	CISI
32	1295	68	CRAN
33	20	950	MED
Doğru Sınıflandırılan:			3636
Yanlış Sınıflandırılan:			194
Başarım Oranı (%) :			94,935

Naive bayes algoritması ile yapılan sınıflandırma neticesinde verilerin farklı sınıflara göre dağılım durumları Şekil 4.3'te gösterilmiştir.



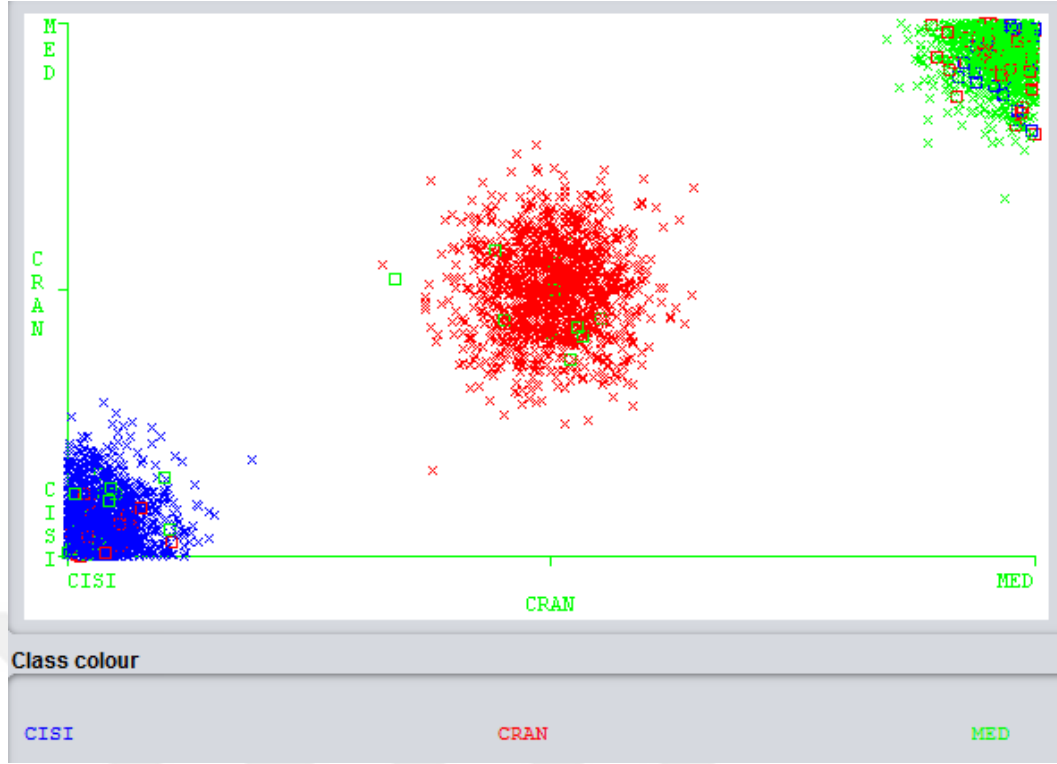
Şekil 4.3. Naive bayes algoritmasının veri dağılımı

#### 4.1.7 Destek Vektör Makineleri (Support Vector Machines)

Tablo 4.5. Destek vektör makineleri algoritmasının başarımı

CISI	CRAN	MED	
1400	0	32	CISI
31	1306	58	CRAN
32	11	960	MED
Doğru Sınıflandırılan:			3666
Yanlış Sınıflandırılan:			164
Başarım Oranı (%) :			95,718

Destek vektör makineleri ile yapılan sınıflandırma neticesinde verilerin farklı sınıflara göre dağılım durumları Şekil 4.4'te gösterilmiştir.



Şekil 4.4. Destek vektör makineleri algoritmasının veri dağılımı

#### 4.1.8 Deneysel Sonuçlarının Değerlendirilmesi

WEKA programında örnek hazır veri seti üzerinde yapılan farklı sınıflandırma algoritmalarından 'Destek Vektör Makineleri' yöntemi en yüksek başarı oranını sağlamıştır.



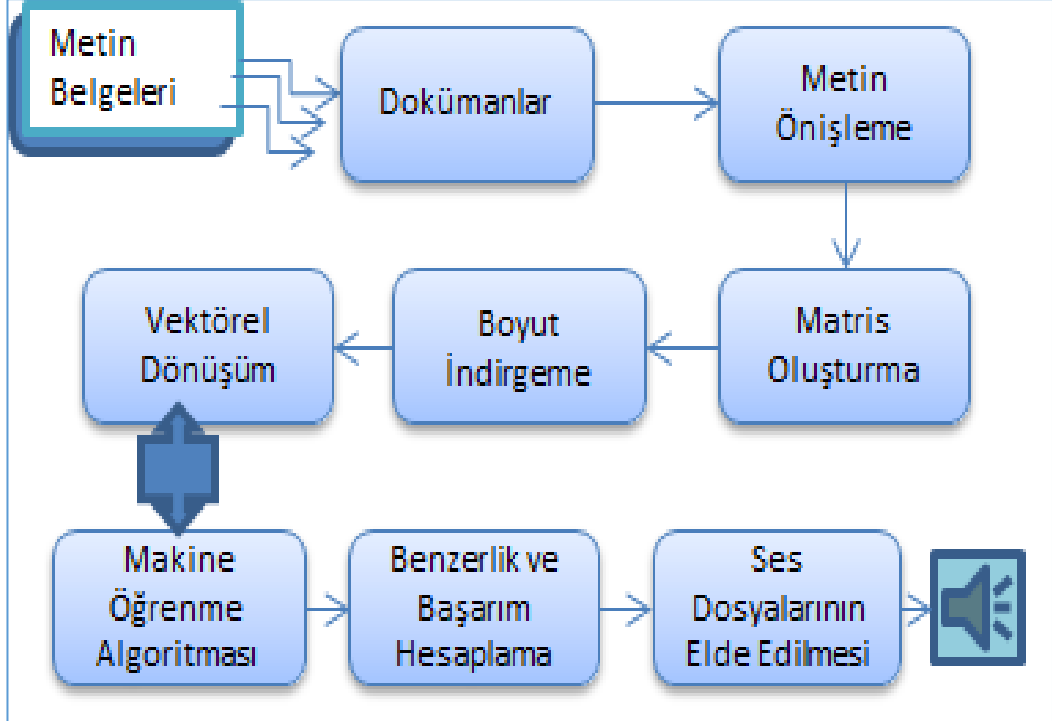
## 5 UYGULAMA GERÇEKLEME

### 5.1 Uygulamanın Amacı

Yapılan uygulamanın amacı, ses sentezleme için insan kulağına doğal gelebilecek bir konuşma elde edebilmek için sentezlenecek metinlerin kategorilere ayrılması (sınıflandırılması), her kategorinin önceden belirlenen kurallar çerçevesinde seslendirilerek doğru sınıflandırılıp sınıflandırılmadığının gözlemlenmesi ve toplam başarı oranının hesaplanmasıdır.

### 5.2 Uygulama Süreci

Uygulama süreci aşağıdaki gibi sıralanmıştır.



Şekil 5.1. Metinden ses sentezleme uygulaması süreçleri

### 5.3 Uygulamannın Gerçekleřtirimi

Uygulama “Windows” iřletim sistemi platformunda, “Java” programlama dili ile “NetBeans” Java derleyici aracı kullanılarak gerekleřtirilmiřtir. Uygulama iin literatürde “classic3” veri seti olarak adlandırılan akıřkanlar dinamięi, bilimsel endeksler ve tıbbi konulardaki 3 farklı türe ait 3891 tane makale özetlerinden oluřan veri seti kullanılmıřtır.

Tablo 5.1. Uygulama veri kümesi tablosu

Doküman Tipi	Sayısı	Konusu
CISI	1460	Akıřkanlar Dinamięi Makale Özetleri
CRAN	1398	Bilimsel Makale Özetleri
MED	1033	Tıbbi Makale Özetleri

Veri kümesi ařaęıdaki ařamalardan geirilerek sınıflandırma iřlemi gerekleřtirilmiř ve bařarım oranı ölçölmüřtür.

#### 5.3.1 Gereksiz Kelimelerin Temizlenmesi

Zamirler, edatlar ve baęlalar metinlerin birbirleriyle karřılařtırılmasında ayrıřtırıcı özellięe sahip olmadıklarından bunların belirlenip temizlenmesi gerekmektedir. Bu iřlem iin “stop-words removal” (Boiy, E. ve ark., 2007) algoritması kullanılmıřtır. Ayrıca tüm kelimeler küçük harf řeklinde tek satırlar haline dönüřtürölmüřtür. Aynı zamanda, noktalama ve rakamsal verilerin sınıflandırma iin herhangi bir öneme sahip olmadıklarından, ilgili dokümanlardan ıkartılması yapılmıřtır. Bu iřlemlerden sonra Tablo 5.2’de verilen bir örnekleme dosyası Tablo 5.3’teki gibi elde edilmiřtir.

Tablo 5.2. Temizleme iřlemi yapılmamıř metin dosyasının örneęi

The relationships between the organization and control of writings and the organization and control of knowledge and information will inevitably enter our story, for writings contain, along with much else, a great deal of mankind's stock of knowledge and information.

Tablo 5.3. Temizleme işlemi yapılmış metin dosyası

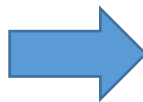
```
relationships
organization
control
writings
organization
control
knowledge
information
```

### 5.3.2 Kelime Köklerinin Tespiti

Metinler içinde geçen kelimeler aynı anlam içermesine rağmen cümle içerisinde farklı ekler alabildiklerinden aynı kökten gelen kelimelerin tespit edilmesi gerekmektedir. Bu işlem için “Porter-Stemming” (Porter M. F., 1980) algoritması kullanılmış ve yukarıdaki stop-words temizleme aşamasından elde edilen örnek dosya Tablo 5.4’teki gibi işlenmiştir.

Tablo 5.4. Kelime kökleri bulunan metin dosyası

```
relationships
organization
control
writings
organization
control
knowledge
information
```



```
relationship
organ
control
write
organ
control
knowledg
inform
```

### 5.3.3 Doküman-Terim Matrisinin Elde Edilmesi

Bu aşamada tüm dokümanların isimleri satırlarda, tüm kelimeler de sütunlarda olacak şekilde bir matris oluşturulmuştur. Tüm kelimelerin hangi dokümanda kaç defa geçtiği hesaplanıp bu matriste gösterilmiş ve son satırda da her kelimenin tüm dokümanlarda kaç defa geçtiği hesaplanmıştır. Bu aşama sonunda 3892 satırdan ve yaklaşık 29000 sütundan oluşan bir matris elde edilmiştir. Aşağıdaki tabloda bunun kısa bir örnek gösterimi verilmiştir.

Tablo 5.5. Doküman-Kelime matrisi

	edit	dewey	decim	classif	present	studi	histori	...
cisi.000001	4	3	2	1	1	1	2	...
cisi.000002	1	0	0	0	1	0	0	...
cisi.000003	0	0	1	0	2	1	0	...
cisi.000004	0	0	0	0	1	2	1	...
...	...	...	...	...	...	...	...	...
<b>Toplam</b>	<b>55</b>	<b>17</b>	<b>26</b>	<b>228</b>	<b>1234</b>	<b>1267</b>	<b>98</b>	...

#### 5.3.4 Boyut Azaltma

Boyut azaltma, dokümanları birbiri ile karşılaştırma ve benzerliklerinin bulunabilmesi için önemli bir adımdır. Boyut azaltma sayesinde daha kesin sonuçlar elde edilebilecek bir alt küme oluşturulur. Tüm dokümanlar içerisinde sadece bir defa geçen bir kelime kıyaslama yapmaya gerek olmayan bir kelime olduğundan bu ve benzer kelimelerin çıkarılması daha doğru sonuç elde edilmesini sağlayacaktır. Bu aşamada tüm dokümanlar içerisinde toplam 10 ve 10'dan daha az geçen kelimelerin ayırt edici özelliği bulunmadığı hesaplanarak (Şekil 5.3) bu kelimeler ilgili matristen çıkarılmıştır. Bu sayede sütun sayısının boyutu azaltılmıştır. 3892 satır (doküman) ve yaklaşık 3400 sütun (kelime) içeren yeni bir matris elde edilmiştir. Her bir satır içerisinde geçen kelimeler bir vektör haline dönüştürülmüştür. Bu işlem dokümanların benzerliklerinin hesaplanabilmesi için gerekli bir işlemdir.

#### 5.3.5 Benzerliklerin Bulunması ve Başarım Oranının Hesaplanması

Bu aşamada öncelikle ayrı ayrı vektörel olarak ifade edilen dokümanlar (satırlar), kümeleme işlemi ile eğitim ve test kümesi şeklinde gruplanmıştır. Tüm doküman kümesinin ilk beşte birlik bölümü eğitim kümesi, geri kalanlar test kümesi olacak şekilde seçilmiştir. Eğitim kümesinin her elemanı test kümesinin her elemanı ile hesaplamaya dâhil edilerek 5-kat çapraz doğrulama işlemi yapılmıştır. Daha sonra benzerlikler hesaplanmıştır. Benzerliklerin hesaplanması için “Kosinüs Benzerliği”

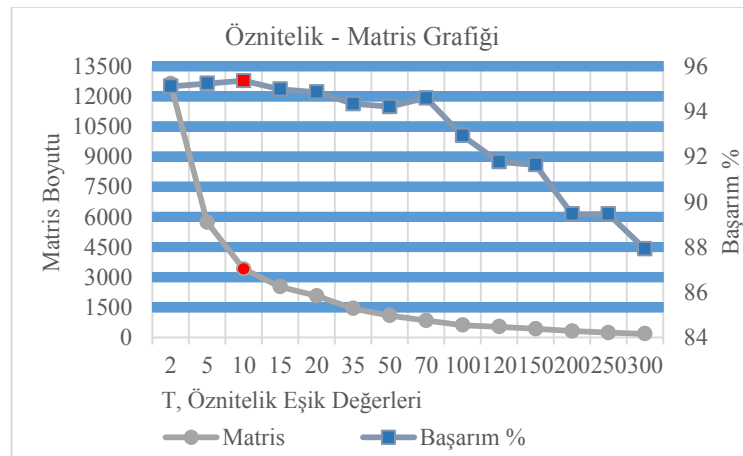
formülünden yararlanılmıştır. Bu aşamada sınıflandırma için gözetimli makine öğrenmesi algoritmalarından k-en yakın komşu algoritmasından yararlanılmıştır.

Tablo 5.6. Benzerliklerin ve başarımın hesaplanması

cisi.000001.txt dosyasına en benzer dosya : cisi.001205.txt Benzerlik oranı : 0.269
cisi.000002.txt dosyasına en benzer dosya : cisi.000916.txt Benzerlik oranı : 0.513
cisi.000003.txt dosyasına en benzer dosya : cisi.001316.txt Benzerlik oranı : 0.321
cisi.000004.txt dosyasına en benzer dosya : cisi.001401.txt Benzerlik oranı : 0.993
.....
Doğru sınıflandırılan doküman sayısı : 742
Yanlış sınıflandırılan doküman sayısı : 36
<b>Başarım Oranı (%) : 95.373</b>

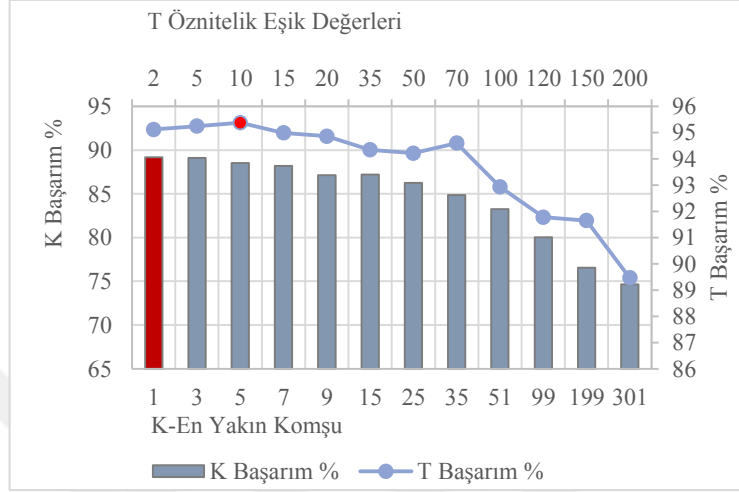
### 5.3.6 Sınıflandırma Başarımının Değerlendirilmesi

Yapılan uygulamada aşağıdaki Şekil 5.3'te görüldüğü gibi tüm doküman kümesinde toplamda geçen kelime sayısı 'T' ile ifade edilecek şekilde eşik değerleri belirlenmiştir. Bu eşik değerinin altında kalanların ayırt edici özelliği bulunmadığı varsayılarak matristen çıkarılmış ve matrisin boyutu azaltılmıştır. Uygulamada 10 eşik değerinde en yüksek başarıma ulaşıldığı görülmüştür. Matris boyutu daha da azaltıldığında ise başarım da düşmüştür (Şirin Y. ve Kutlugün M.A., 2017).



Şekil 5.2. Boyut azaltmanın başarıma etkisi

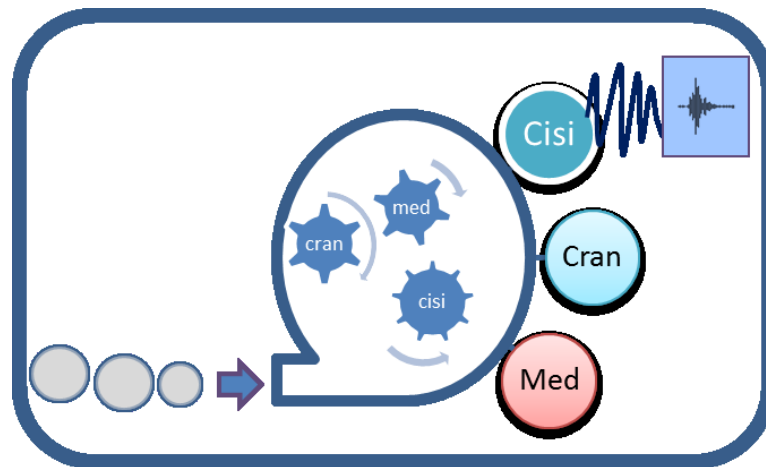
Uygulama farklı k değerleri için de uygulanmış, bu veri seti için en yüksek başarımları oranı yüzde 95,373 olarak k=1 değerinde elde edilmiştir. Bu veri kümesinde k değeri arttıkça sınıflandırma başarımları oranı azaldığından k=1 değeri ile en yüksek başarımları ulaşılmıştır (Şirin Y. ve Kutlugün M.A., 2017).



Şekil 5.3. Farklı k değerlerinin başarımları etkisi

### 5.3.7 Ses Dosyalarının Elde Edilmesi

Ses dosyalarının elde edilmesi için “NetBeans” Java derleyici aracına “FreeTTS” kütüphaneleri eklenmiş ve “MBROLA” ses veri tabanı tanımlanmıştır. Başlangıçta 3 türe ait aşağıdaki varsayılan ses biçimleri aşağıdaki tablolardaki gibi tanımlanmıştır. Bu değerler doğru sınıflandırılan dokümanlarda kullanılarak ses dosyaları “.wav” ses biçiminde elde edilmiştir. Doğru seslendirme süreci şekilsel olarak aşağıdaki gibidir.



Şekil 5.4. Doğru seslendirme süreci

Tablo 5.7. CISI türündeki dosyaların ses özellikleri

Ses Dosyası	:	mbrola_us1 - 16kHz
Cinsiyet	:	Bayan Sesi
Okuma Hız Değeri	:	1.0f
Perde Frekans Değeri	:	180f
Perde Frekans Aralığı	:	22.0f
Ses Yoğunluğu	:	1.0f

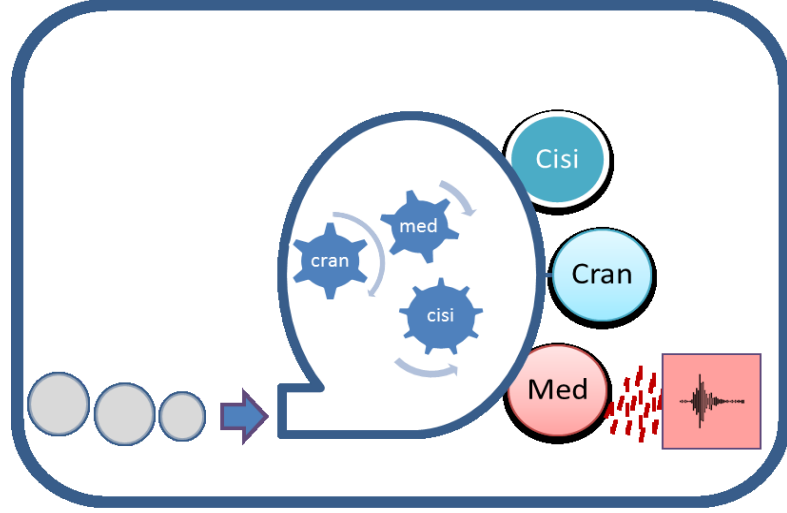
Tablo 5.8. CRAN türündeki dosyaların ses özellikleri

Ses Dosyası	:	kevin16 - 16kHz
Cinsiyet	:	Erkek sesi
Okuma Hız Değeri	:	1.0f
Perde Frekans Değeri	:	100f
Perde Frekans Aralığı	:	11.0f
Ses Yoğunluğu	:	1.0f

Tablo 5.9. MED türündeki dosyaların ses özellikleri

Ses Dosyası	:	mbrola_us2 - 16kHz
Cinsiyet	:	Erkek sesi
Okuma Hız Değeri	:	1.0f
Perde Frekans Değeri	:	115f
Perde Frekans Aralığı	:	12.0f
Ses Yoğunluğu	:	1.0f

Yanlış sınıflandırılan dokümanlar için, kendi kategorisi haricinde, sınıflandırıldığı türe göre ses dosyaları oluşturulmuştur. Yanlış veya hatalı seslendirme süreci şekilsel olarak Şekil 5.5'teki gibidir.



Şekil 5.5. Hatalı seslendirme süreci

Süreç sonunda elde edilen ses dosyaları incelendiğinde ses tonlamalarındaki farklılıklardan hangi dokümanların doğru, hangi dokümanların yanlış veya hatalı sınıflandırıldığı açıkça tespit edilebilmektedir.



## 6 SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında ses sentezleme konusunun metinden ses sentezleme alanı ile ilgili detaylı bilgilere yer verilmiş, makine öğrenmesi algoritmaları yardımıyla ses sentezleme konusu uygulamalı olarak ele alınmıştır.

Bu çalışmada insan doğasına uygun alışlagelmiş seslendirmeler yapabilmek için veri türlerinin sınıflandırılması gerektiği tezi üzerinde ağırlıklı olarak durulmuş, ses veri tabanının hangi algoritmalar ile elde edildiği ve nasıl birleştirildiği kapsam dışında tutulmuştur. Bu işlemler için “FreeTTS” açık kaynak kodlu Java programlama dili kütüphaneleri ve “MBROLA” ses veri tabanı kullanılmıştır.

Metinden ses sentezlemenin temel süreçlerinden biri olan metin önışleme aşamasında metinlerin sınıflandırılması ve bu sayede metinler hakkında detaylı bilgiler elde edilerek hangi metin türlerinin ne şekilde seslendirileceği konusunda önemli sonuçlara ulaşılmıştır. Bu işlemler sonunda doğru sınıflandırılan dokümanların ses dosyaları varsayılan olarak belirlenmiş düzgün tonlamalar ile elde edilirken, yanlış sınıflandırılan dokümanlar için kendi kategorisi dışında farklı sesler elde edilmiştir.

Metinler sınıflandırılırken kullanılan algoritmaların tutarlılığı ve geniş bir veri ambarına sahip olunması başarıyı etkileyen önemli unsurlardandır.

Problem kümesinden sayısal hesaplamalar yapmak üzere bazı çıkarımlarda bulunabilmek için metinlerin işlenerek vektörel olarak ifade edilmesi zorunludur. Bu işlem, önce ayırt edici unsurların doğru kriterler ile belirlenerek gereksiz görülen ve gürültülü veri olarak adlandırılan verilerin bu kümeden çıkarılması ile mümkündür. Bu şekilde metin üzerinde gerekli işlemler yapıldıktan sonra makine öğrenmesi algoritmaları uygulanmalıdır. İlave olarak gürültülü verilerin tespiti için gerekirse metni cümlelere ayırma yoluyla ayırt edici bilgi içermeyen veya yanlış yönlendirmelere sebep olabilecek cümleler metin sınıflandırma aşamasından önce metinden atılmalıdır.

Ayrıca hazır veri seti üzerinde “WEKA” aracı ile tanımlı gelen makine öğrenmesi algoritmalarını karşılaştırma amacıyla kullanmanın yanında, Java program kodları ile başka bir veri kümesindeki işlenmemiş metinler belirli işlemlerden geçirilerek hesaplanan değerlere göre seçilen makine öğrenmesi algoritması uygulanmıştır.

Bu çalışma gelecek çalışmalar için gerek makine öğrenmesi ve veri madenciliği, gerek ses sentezleme alanlarına temel teşkil edecek niteliktedir. Bu çalışmalara daha fazla katkı sağlayabilmek için farklı veri temizleme ve kelime kök bulma algoritmaları kullanılması önerilir. Ayrıca test ve eğitim kümelerinin farklı boyutlarda belirlenerek 5 kat veya 10 kat çapraz geçirme yöntemleri ile uygulanması başarıyı artırıcı unsurlar olacaktır.

Tek bir düz metin içinde farklı kategorilerdeki metin türlerinin birlikte verildiği durumlar için paragraf paragraf sınıflandırma işlemlerinin yapılarak seslendirilmesinin daha olumlu sonuçlar vereceği düşünülmektedir.

Bu uygulama çerçevesinde kullanılan işletim sistemi platformu ve seçilen ses kütüphanelerinin uyumsuzluğu ile ilgili bazı zorluklar ile karşılaşmıştır. Bunların sebepleri araştırılarak giderilmiş ve ses dosyaları başarılı bir şekilde elde edilebilmiştir. Daha sonraki çalışmalarda bu tür zorluklar ile karşılaşmamak için birbiriyle uyumlu olan platform ve kütüphanelerin kullanılması önerilir.

## KAYNAKÇA

- Aksan D. (2000). *Türkiye Türkçesinin Dünü, Bugünü, Yarını*. İstanbul: Bilgi Yayınları.
- Allen, J. ve ark. (1987). *From Text to Speech: The MITalk System*. Cambridge University Press.
- Alpaydın, E. (2004). *Introduction to Machine Learning*. The MIT Press, Sayfa: 3-6.
- Alzand, H.R.A. ve Karacan H. (2014). Bölümleyici kümeleme algoritmalarının farklı veri yoğunluklarında karşılaştırılması. *ISSN 1012-2354* (s. 56-62). Kayseri: Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi.
- Artuner, H. (1994). *Bir Türkçe Fonem Kümeleme Sistemi Tasarımı ve gerçekleştirimi, Doktora Tezi*. Ankara: Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü.
- Aşlıyan R. ve ark. (2006). Türkçe Otomatik Heceleme Sistemi ve Hece İstatistikleri. *Akademik Bilişim 2006 BilgiTek IV*. Denizli: Pamukkale Üniversitesi.
- Aşlıyan R. ve Günel K. (2008). Türkçe Metinler için Hece Tabanlı Konuşma Sentezleme Sistemi. Çanakkale: Çanakkale Onsekiz Mart Üniversitesi, Akademik Bilişim.
- Aşlıyan, R., ve Günel, K. (2005). *Design and Implementation For Extracting Turkish Syllables And Analysing Turkish Syllables*”,. İstanbul: INISTA-International Symposium on Innovations in Intelligent Systems and Applications, Yıldız Technical University.
- Aydemir T. ve Yılmaz, A. E. (2010). Türkçe fiil çekimlerinde vurgu konumunu belirlemek için bir yazılım kütüphanesi. (s. s. 696–699). Diyarbakır: IEEE 18. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU 2010).
- Aydın, Ö. (2005). *Yapay sinir ağlarını kullanarak bir ses tanıma sistemi geliştirilmesi Yüksek Lisans Tezi*. Edirne: Trakya Üniversitesi Fen Bilimleri Enstitüsü.
- Ayhan, K. (1998). *Text to Speech Synthesizer in Turkish Using Non Parametric Techniques*. ODTÜ, Yüksek Lisans Tezi.
- Bachan, J. (2010). *Efficient Diphone Database Creation for MBROLA a Multilingual Speech Synthesizer*. Adam Mickiewicz University : XII International PhD Workshop.
- Biricik, G. (2011). *Metin Sınıflama İçin Yeni Bir Özellik Çıkarım Yöntemi*. İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi.
- Boiy, E. ve ark. (2007). *Automatic Sentiment Analysis in On-line Text*. Openness in Digital Publishing: Awareness, Discovery and Access In : Proceedings of the 11th International Conference on Electronic Publishing. ELPUB2007.
- Can, B. (2007). *Bir hece-tabanlı Türkçe sesli ifade tanıma sisteminin tasarımı ve gerçekleştirimi, Yüksek Lisans Tezi*. Ankara: Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü.
- Canal Ş.M. ve ark. (2010). Türkçe Metinden Konuşma Sentezlemede Yaşanan Sıkıntılar ve Çözüm Yöntemleri. *Havacılık ve Uzay Teknolojileri Dergisi*.

- Chao, W. (2011). *Machine Learning Tutorial*. Taiwan: National Taiwan University, Graduate Institute of Communication Engineering, DISP Lab.
- Chen, J. ve ark. (2009). A Fast k-means Clustering Algorithm Based on Grid Data Reduction. *9980042(1095-323X)* (s. 1-6). IEEE digital library.
- Cohen W. W. ve Singer Y. (1996). *Context-Sensitive Learning Methods for Text Categorization*. Proceedings of the 19th Annual ACM SIGIR Conference.
- Cover, T.M. ve Hart, P.E. (1967 ). *Nearest Neighbor Pattern Classification*. IEEE Transactions on Information Theory, 13:21-27.
- Çoban, Ö. (2016). *Metin Sınıflandırma Teknikleri ile Türkçe Twitter Duygu Analizi*. Erzurum: Atatürk Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi.
- Dasgupta, A. ve ark. (2007). *Feature selection methods for text classification*. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 230-239). ACM.
- Eker, B. (2002). *Turkish Text to Speech System, Yüksek Lisans Tezi*. Ankara: Bilkent Üniversitesi, Mühendislik ve Fen Bilimleri Enstitüsü.
- Erdemir C. (2010). *Türkçe Metin Seslendirme İçin Doğal Konuşma Sentezlem*. İstanbul: İstanbul Üniversitesi, Yüksek Lisans Tezi.
- Ergenç, İ. (2002). *Spoken Language and Dictionary of Turkish Articulation*. İstanbul: Multilingual Yabancı Dil Yayınları.
- Fieldman, R. and Sanger J. . (2006). *The text mining handbook advanced approaches in advanced ana-lyzing unstructured data*. Cambridge University Press.
- Flanagan J. L. ve ark. (2008). *Speech Analysis Synthesis and Perception*. New York: Springer.
- Forman, G. (2003). *An extensive empirical study of feature selection metrics for text classification*. The Journal of machine learning research, 3, 1289-1305.
- Gebremariam, G. (2016). *Speech synthesis, Thesis*. Helsinki : Helsinki Metropolia University of Applied Sciences.
- Guyon, I. ve Elisseeff, A. (2003). *An introduction to variable and feature selection*. The Journal of Machine Learning Research, 3, 1157-1182.
- Güldalı, K. (2009). *Türkçe Metin Seslendime Yüksek Lisans Tezi*. İstanbul: İstanbul Teknik Üniversitesi.
- Haykin, S. (1994). *Neural Networks*. USA: Macmillan College Publishing Company, 696,.
- Herbrich, R. (2002). *Learning Kernel Classifiers*. The MIT Press, ISBN 0-262-08306-X.
- Hinton G. ve Sejnowski T.J. (1999). *Hinton G. ve Sejnowski T.J. (editorler), 1999, "Unsupervised Learning and Map Formation: Foundations of Neural Computation", MIT Press, ISBN 0-262-58168-X*. MIT Press, ISBN 0-262-58168-X.
- Jiawei, H. (2006). *Cluster Analysis, Data Mining: Concepts and Techniques*. U.S.A.: Elsevier Inc.

- Jurafsky D., J. H. (2008). *Speech and Language Processing*. Prentice Hall.
- Kecman, V. (2001). *Learning and Soft Computing: Support Vector Machine, Neural Networks, and Fuzzy Logic Models*. The MIT Press, ISBN 0-262-11255-8.
- Kılınç D. ve ark. (2015). *Metin Madenciliği Kullanılarak Yazılım Kullanımına Dair Bulguların Elde Edilmesi*. 9. Ulusal Yazılım Mühendisliği Sempozyumu, Yaşar Üniversitesi.
- Klatt, D. H. (1987). Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, Cilt 82, 737 – 793.
- Kotsiantis, S. B. (2011). *Supervised machine learning: A review of classification techniques*. Greece: University of Peloponnese, Department of Computer Science and Technology.
- Kutlugün ve ark. (2017). *Yapay Sinir Ağları ve K-En Yakın Komşu Algoritmalarının Birlikte Çalışma Tekniği (Ensemble) ile Metin Türü Tanıma*. İstanbul: XXII. Türkiye’de İnternet Konferansı (inet’tr17), Bahçeşehir Üniversitesi,.
- Külekçi M. O. ve Oflazer K. (2006). An infrastructure for Turkish prosody generation in text-to-speech synthesis. (s. s. 49–57). Muğla: 15th Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN 2006.
- Lemmetty, S. (1999). Review of Speech Synthesis Technology. *Helsinki University of Technology, Yüksek Lisans tezi*. Helsinki.
- Li, Y. H. ve Jain, A. K. (1998). *Classification of text documents*. The Computer Journal.
- Ljungqvist M. ve ark. (1994). *A New System for text-to-Speech and Its Applications to Swedish*. ICSLP94 (4) : 1779-1782,1994.
- Mahwash, A. ve Shibli, N. (2014). Text-to-Speech Synthesis using Phoneme Concatenation. *ISSN : 2277-1581* (s. Volume No.3 Issue No.2, pp : 193 – 197). International Journal of Scientific Engineering and Technology.
- Manning, D. C. ve Schütze H. (1999). *Foundations of Statistical Natural Language Processing*. Massachusetts: The MIT Press, Cambridge, Massachusetts.
- Mönius B. ve ark. (1995). *Recent Advances Multilingual Text-to-Speech Synthesis*. Fortschritte der Akustik - DAGA.
- Nizam, H. ve ark. (2014). *Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Dengeli ve Dengesiz Veri Setlerinin Performanslarının Karşılaştırılması*. XIX. Türkiye’de İnternet Konferansı.
- Oflazer K. (1994). *Two-level description of Turkish morphology*. Literary and Linguistic Computing, 9, 175-198 .
- Oskay B. ve ark. (2001). Türkçe metinden konuşma sentezlemede ezgi belirlenmesi ve uygulanması. (s. s. 238–243). IEEE 9. Sinyal İşleme ve Uygulamaları Kurultayı SİU-2001.
- Özen, S. S. (2002). *Türkçe Metinden Konuşma Sentezleme*. Hacettepe Üniversitesi Yüksek Lisans Tezi.
- Özgür, A. (2004). *Supervised and unsupervised machine learning techniques for text document categorization*. İstanbul: Boğaziçi Üniversitesi, Yüksek Lisans Tezi.

- Özkan, Y. (2008). Karar Ağaçları ile Sınıflandırma. D. R. Uğutkaya içinde, *Veri Madenciliği Yöntemleri* (s. 216). İstanbul: Papatya Yayıncılık.
- Öztemel E. (2006). *Yapay Sinir Ağları*. İstanbul: Papatya yayıncılık, 2. baskı, .
- Öztürk Ö. (2005). *Modelling phoneme durations and fundamental frequency contours in Turkish speech*. ODTÜ Fen Bilimleri Enstitüsü, Doktora Tezi,.
- Patra, A. ve Singh, D. (2013). *A Survey Report on Text Classification with Different Term Weighing Methods and Comparison between Classification Algorithms*. International Journal of Computer Applications, 75(7).
- Pehlivan, R. (2014). *Resim Tabanlı Osmanlıca Belgelerde Sınıflandırma*. İstanbul: İstanbul Kültür Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Pilavcılar, İ. (2007). *Metin Madenciliği İle Metin Sınıflandırma*. İstanbul: Yıldız Teknik Üniversitesi Matematik Mühendisliği, Yüksek Lisans Tezi.
- Porter M. F. (1980). *An Algorithm for Suffix Stripping*. Program, Vol. 14, syf.
- Rabiner L. R., Juang B. H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall.
- Rosenblatt, F. (1959). *Principles of Neuradynamics*. New York: Spartan Books.
- Rosenblatt, F. (1962). *Principles of Neuradynamics: Perceptrons and The Theory of Brain Mechanisms*. Washington D.C.: Spartan Boks.
- Salton G. ve ark. (1975). *A Vector-Space Model for Automatic Indexing*. Communications of the ACM, Vol. 18, No. 11, syf. 613-620.
- Salton, G. ve Buckley, C. . (1988). *Term-weighting approaches in automatic text retrieval*. Information processing & management, 24(5), 513-523 .
- Sel İ. ve ark. (2011). *Beyin Bilgisayar Arayüzleri İçin Türkçe Metinden Konuşma Sentezleme Sistemi*. Elazığ: Fırat Üniversitesi Elektrik-Elektronik Bilgisayar Sempozyumu, FEEB-2011,.
- Sel, İ. (2013). *Türkçe Metinler İçin Hece Tabanlı Metinden Konuşma Sentezleme Sistemi Yüksek Lisans Tezi*. Elazığ: Fırat Üniversitesi.
- Şayli Ö. (2002). *Duration analysis and modelling for Turkish text-to-speech synthesis*. Boğaziçi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Şen K.Ö. ve ark. (2007). Bir Konuşmacının Konuştuğu Dilin Belirlenmesi. *III. İletişim Teknolojileri Ulusal Sempozyumu*. Adana.
- Şentürk T. ve Adalı E. (2010). Türkçe Metin Seslendirme. *BBM Dergisi*, 4, 35-51.
- Şentürk, T. (2010). *Türkçe Metin Seslendirme Yüksek Lisans Tezi*. İstanbul: İstanbul Teknik Üniversitesi.
- Şirin Y. ve Kutlugün M.A. (2017). Konuşma Sentezinde Artan Doğallık için Boyut Azaltma Seçimi. *Signal Processing and Communications Applications Conference (SIU), 2017 25th* (s. 10.1109/SIU.2017.7960573). Antalya: IEEE.
- Taşova, O. (2011). *Yapay Sinir Ağları ile Yüz Tanıma*. İzmir: Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Tekeli K. ve Aşliyan R. (2016). *Çok Katmanlı Algılayıcı, K-NN ve C4.5 Metotlarıyla İstenmeyen E-postaların Tespiti*. Aydın: XVIII. Akademik Bilişim Konferansı.

- Tekindal B. ve Arık G. (2012). Görme Engelliler için Türkçe Metinden Konuşma Sentezleme Yazılımı Geliştirilmesi. (s. CİLT: 5, SAYI: 3,). Bilişim Teknolojileri Dergisi.
- Telcioğlu, M. (2007). *Veri Madenciliğinde Genetik Programlama Temelli Yeni Bir Sınıflandırma Yaklaşımı ve Uygulaması*. Kayseri: Erciyes Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi.
- Therrien, C. W. (1989). *Decision estimation and classification: an introduction to pattern recognition and related topics*. Monterey, CA: Naval Postgraduate School, ISBN:0-471-83102-6.
- Uslu İ.B. ve ark. (2011). Türkçe metinden konuşma sentezlemede fiil çekimleri için yeni bir ezgi modeli. (s. s. 638–641). Antalya: IEEE 19. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, SIU-2011.
- Uslu, İ. B. (2010). *Metinden Konuşma Sentezleme*. Ankara: TMMOB Elektrik Mühendisleri Odası.
- Uzun, E. (2007). *İnternet Tabanlı Bilgi Erişimi Destekli Bir Otomatik Öğrenme Sistemi*. Edirne: Trakya Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi.
- Üçok, N. (1951). *Genel Fonetik*. Ankara Üniversitesi Dil, Tarih ve Coğrafya Fakültesi Yayınları.
- Ünalı, İ. (2007). *Taşınabilir cihazlar için Türkçe metinden konuşma sentezleme sistemi, Yüksek Lisans Tezi*. Ankara: Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü.
- van Rijsbergen C. J. (1979). *Information Retrieval*. London: Butterworths.
- Verhelst, W. and Roelands, M. (1993). Speech, and Signal Processing. *Proceedings for the International Conference of Acoustics*. Minneapolis.
- Vural, E. (2003). *A Prosodic Turkish Text-to-Speech Synthesizer*. Sabancı Üniversitesi, Mühendislik ve Doğa Bilimleri Enstitüsü, Yüksek Lisans Tezi,.
- Wu, C.H. ve Tsai, C.H. (2009). *Robust classification for spam filtering by back-propagation neural networks using behavior-based features*. Applied Intelligence, 31(2), 107-121.
- Yang, Y. ve Pedersen, J. O. (1997). *A comparative study on feature selection in text categorization*. In ICML (Vol. 97, pp. 412-420).
- Yaşaroğlu, Y. (2003). *Multi-Model Video Summarization Using Hidden Markov Models for Content-Based Multimedia Indexing*. Ankara: Middle East Technical University.
- Yılmaz, A. E. (2009). Türkçe Metinden Konuşma Sentezleme Uygulamaları için bir Veri Sözlük Seti ve Yazılım Çerçevesi Önerisi. 17. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (s. SIU-2009). Antalya: IEEE.
- Yılmaz, R. (2013). *Türkçe Dokümanların Sınıflandırılması, Yüksek Lisans Tezi*. Aydın: Adnan Menderes Üniversitesi.

## EKLER

Bu bölümde uygulama gerçekleştirme bölümünde kullanılan bazı önemli program kod parçaları gerekli açıklamaları ile aşağıda sunulmuştur.

- Stop-Words temizleme işlemleri için kullanılan kodlar;

```
import java.io.*;
import java.util.Scanner;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

public class kopya {

String[] stopWrds = {"without", "see", "unless", "due",
"also", "must", "might", "like", "]", "[", "}", "{", "<", ">",
"?", "\"", "\\", "/", ")" , "(" , "will", "may", "can", "much",
"every", "the", "in", "other", "this", "the", "many", "any",
"an", "or", "for", "in", "an", "an ", "is", "a", "about",
"above", "after", "again", "against", "all", "am", "an",
"and", "any", "are", "aren't", "as", "at", "be", "because",
"been", "before", "being", "below", "between", "both", "but",
"by", "can't", "cannot", "could", "couldn't", "did", "didn't",
"do", "does", "doesn't", "doing", "don't", "down", "during",
"each", "few", "for", "from", "further", "had", "hadn't",
"has", "hasn't", "have", "haven't", "having", "he", "he'd",
"he'll", "he's", "her", "here", "here's", "hers", "herself",
"him", "himself", "his", "how", "how's", "i ", " i", "i'd",
"i'll", "i'm", "i've", "if", "in", "into", "is",
"isn't", "it", "it's", "its", "itself", "let's", "me", "more",
"most", "mustn't", "my", "myself", "no", "nor", "not", "of",
"off", "on", "once", "only", "ought", "our", "ours",
"ourselves", "out", "over", "own", "same", "shan't", "she",
"she'd", "she'll", "she's", "should", "shouldn't", "so",
"some", "such", "than", "that", "that's", "their", "theirs",
"them", "themselves", "then", "there", "there's", "these",
"they", "they'd", "they'll", "they're", "they've", "this",
"those", "through", "to", "too", "under", "until", "up",
"very", "was", "wasn't", "we", "we'd", "we'll", "we're",
"we've", "were", "weren't", "what", "what's", "when",
"when's", "where", "where's", "which", "while", "who",
"who's", "whom", "why", "why's", "with", "won't", "would",
```



```

"wouldn't", "you", "you'd", "you'll", "you're", "you've",
"your", "yours", "yourself", "yourselves", "Without", "See",
"Unless", "Due", "Also", "Must", "Might", "Like", "Will",
"May", "Can", "Much", "Every", "The", "In", "Other", "This",
"The", "Many", "Any", "An", "Or", "For", "In", "An", "An ",
"Is", "A", "About", "Above", "After", "Again", "Against",
"All", "Am", "An", "And", "Any", "Are", "Aren't", "As", "At",
"Be", "Because", "Been", "Before", "Being", "Below",
"Between", "Both", "But", "By", "Can't", "Cannot", "Could",
"Couldn't", "Did", "Didn't", "Do", "Does", "Doesn't", "Doing",
"Don't", "Down", "During", "Each", "Few", "For", "From",
"Further", "Had", "Hadn't", "Has", "Hasn't", "Have",
"Haven't", "Having", "He", "He'd", "He'll", "He's", "Her",
"Here", "Here's", "Hers", "Herself", "Him", "Himself", "His",
"How", "How's", "I ", " I", "I'd", "I'll", "I'm", "I've",
"If", "In", "Into", "Is", "Isn't", "It", "It's", "Its",
"Itself", "Let's", "Me", "More", "Most", "Mustn't", "My",
"Myself", "No", "Nor", "Not", "Of", "Off", "On", "Once",
"Only", "Ought", "Our", "Ours", "Ourselves", "Out", "Over",
"Own", "Same", "Shan't", "She", "She'd", "She'll", "She's",
"Should", "Shouldn't", "So", "Some", "Such", "Than",
"That", "That's", "Their", "Theirs", "Them", "Themselves",
"Then", "There", "There's", "These", "They", "They'd",
"They'll", "They're", "They've", "This", "Those", "Through",
"To", "Too", "Under", "Until", "Up", "Very", "Was", "Wasn't",
"We", "We'd", "We'll", "We're", "We've", "Were", "Weren't",
"What", "What's", "When", "When's", "Where", "Where's",
"Which", "While", "Who", "Who's", "Whom", "Why", "Why's",
"With", "Won't", "Would", "Wouldn't", "You", ".", ",", ";",
":", "..", "You'd", "You'll", "You're", "You've", "Your",
"Yours", "Yourself", "Yourselves"};

```

```

public void main(String[] args) {
File srcFolder = new
File("C:\\Speech_Synthesis_DataSet\\classic3");
File destFolder = new
File("C:\\Speech_Synthesis_DataSet\\classic3_temiz");

//Kaynakta klasör var mı yok mu kontrolü
if(!srcFolder.exists()){
System.out.println("Kaynak bulunamadı.");
//Çıkış
System.exit(0);
}else{
try{
copyFolder(srcFolder,destFolder);
}catch(IOException e){
e.printStackTrace();
//Hata oluştu, Çıkış
System.exit(0);
}
}
System.out.println("İşem bitti");
}

public void copyFolder(File src, File dest)

```

```

throws IOException{
if(src.isDirectory()){
//Eğer hedef klasör yoksa, oluştur
if(!dest.exists()){
dest.mkdir();
System.out.println(src + " kopyalanıyor " + dest);
}

//Klasör altındaki bütün dosya ve klasörleri listele
String files[] = src.list();
for (String file : files) {
//Const. Metodlar
File srcFile = new File(src, file);
File destFile = new File(dest, file);
//recursive kopyalama fonksiyon
//copyFolder(srcFile,destFile);

        try {
InputStream in = new FileInputStream(srcFile);
OutputStream out = new FileOutputStream(destFile);
Scanner fipl = new Scanner(srcFile);
OutputStream out2 = new FileOutputStream(destFile);
            while (fipl.hasNext()) {
                int flag = 1;
                String s1 = fipl.next();
                s1 = s1.toLowerCase();
                for (String stopWrd : stopWrds) {
                    if (s1.equals(stopWrd)) {
                        flag = 0;
                    }
                }

                if (flag != 0) {
                    System.out.println(s1);
                    PrintStream p = new PrintStream(out2);
                    p.println(s1);
                }
            }
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}else
//Eğer dosya ise
//Bytes stream ile dosya tipine bakmadan hedefe kopyala
InputStream in = new FileInputStream(src);
OutputStream out = new FileOutputStream(dest);
byte[] buffer = new byte[1024];
int length;
//dosya içeriği bytes lara bölünerek kopyalanıyor
while ((length = in.read(buffer)) > 0){
out.write(buffer, 0, length);
}

in.close();
out.close();
System.out.println("Dosya kaynaktan " + src + ", "
+ dest + " hedefe kopyalandı.");

```

```
}  
}  
}
```

- Kelime Köklerinin Porter Stemming Algoritması ile bulunması

```
import java.io.*;  
import java.util.Scanner;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
  
public class EnglishStemmer_porterstemmer {  
public static void main(String[] args) throws Exception {  
File srcFolder = new  
File("C:\\Speech_Synthesis_DataSet\\classic3_temiz");  
File destFolder = new  
File("C:\\Speech_Synthesis_DataSet\\classic3_temiz_kok");  
String a, s, line, token;  
//Kaynakta klasör varmı yokmu kontrolü  
if(!srcFolder.exists()){  
System.out.println("Kaynak bulunamadı.");  
//Çıkış  
System.exit(0);  
}else{  
copyFolder(srcFolder, destFolder);  
if (srcFolder.isDirectory()) {  
System.out.println(" Kaynak " + srcFolder);  
System.out.println(" Hedef " + destFolder);  
String files[] = srcFolder.list();  
System.out.println(files.length);  
String files2[] = srcFolder.list();  
int d = files2.length;  
System.out.println("----- Toplam " + d + " tane dosya var");  
for (String file : files) {  
File srcFile = new File(srcFolder, file);  
File destFile = new File(destFolder, file);  
System.out.println(srcFile.getName() + " --- " +  
srcFile.length() + " tane harf var");  
System.out.println(file);  
InputStream inn = new FileInputStream(srcFile);  
Scanner fip1 = new Scanner(srcFile);  
FileWriter fstream = new FileWriter(destFile);  
OutputStream outt = new FileOutputStream(destFile);  
byte[] buffer = new byte[5120];  
int length;  
while (fip1.hasNextLine()) {  
line = fip1.nextLine();  
Scanner linesc = new Scanner(line);  
while (linesc.hasNext()) {  
token = linesc.next();  
a=EnglishSnowballStemmerFactory.getInstance().process(token);  
}
```

```

System.out.println(a + " ok");
System.out.println(" " + a.length() + " harf var ");

        // En önemli bölüm : Son halini Hedefe kopyala
        PrintStream p = new PrintStream(outt);
        p.println(a);
    }
}
System.out.println("Her dosya sonuna ulaşıldı");
System.out.println(files.length + " dosya bitti");
}
}
System.out.println("The End");
}
}

public static void copyFolder(File src, File dest)
    throws IOException{
    if(src.isDirectory()){
        //Eğer hedef klasör yoksa, oluştur
        if(!dest.exists()){
            dest.mkdir();
            System.out.println(src + " kopyalanıyor " + dest);
        }
    }
}
}
}

```

#### - Dokümanlarda Sık geçen kelimelerin bulunarak Matris elde edilmesi

```

import java.io.FileNotFoundException;
import java.util.List;
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.StringTokenizer;
import java.io.File;
import java.io.IOException;

public class ReadingMultipleFile {

    public static List keywordList = new ArrayList();
    // Ayıklanmış matrisin sütunlarındaki kelimeler için..
    public static List keywordList2 = new ArrayList();
    public static int[][] countMatrix;
    public static int[][] countMatrix2;
    // sütunlar toplamı için tek boyutlu matrix
    public static int[][] counttemp;
    public static int counter = 0;
    public static int[] cc;
    public static int tt = 0;
    static String path =
"C:\\Speech_Synthesis_DataSet\\classic3_temiz_kok";
    public static File folder = new File(path);
    public static File[] listOfFiles = folder.listFiles();
}

```

```

static File destFolder = new
File("C:\\Speech_Synthesis_DataSet\\classic3\\yanlis");

public static void main(String[] args) throws
FileNotFoundException, IOException {

    if(!destFolder.exists()){
        destFolder.mkdir();
    }

    String s_dizisi, tempdizisi;
    StringTokenizer stokenizer;
    int countKeyword = 0;

    if (listOfFiles.length > 0) {
        for (int i = 0; i < listOfFiles.length; i++) {

            if (listOfFiles[i].isFile()) {
                String p1 = listOfFiles[i].getName();
                BufferedReader br = new BufferedReader(new
FileReader(listOfFiles[i].getPath()));
                while ((s_dizisi = br.readLine()) != null) {
                    stokenizer = new StringTokenizer(s_dizisi, " ", false);
                    while (stokenizer.hasMoreTokens()) {
                        tempdizisi = stokenizer.nextToken();
                        if (tempdizisi.length() <= 3 || tempdizisi.length() >= 15) {
                            if (stokenizer.hasMoreTokens()) {
                                stokenizer.nextToken();
                            }
                        } else {
                            if (keywordList.contains(tempdizisi) == false){

                                keywordList.add(tempdizisi);
                                countKeyword++;
                            }
                        }
                    }
                    if (stokenizer.hasMoreTokens()) {
                        stokenizer.nextToken();
                    }
                } // while sonu
            } // while sonu
        }
    }

    countMatrix=new int[listOfFiles.length+1][keywordList.size()];
    //+1 Sutunlar Toplamı için son satır
    counttemp = new int[1][keywordList.size()];
    // toplam satırını için tek boyutlu matrix

    for (int i = 0; i < listOfFiles.length; i++) {
        for (int j = 0; j < keywordList.size(); j++) {
            countMatrix[i][j] = 0;
        }
    }
}

```

```

    }

    if (listOfFiles.length > 0) {
        for (int i = 0; i < listOfFiles.length; i++) {
            if (listOfFiles[i].isFile()) {
                String pl = listOfFiles[i].getName();

BufferedReader br = new BufferedReader(new
FileReader(listOfFiles[i].getPath()));
                while ((s_dizisi = br.readLine()) != null) {
                    tokenizer = new StringTokenizer(s_dizisi, " ", false);
                    while (tokenizer.hasMoreTokens()) {
                        tempdizisi = tokenizer.nextToken();
if (tokenizer.hasMoreTokens()) {
                    tokenizer.nextToken();
                } else {
                    if (tempdizisi.length() <= 3 || tempdizisi.length() >= 15) {
                        if (tokenizer.hasMoreTokens()) {
                            tokenizer.nextToken();
                        }
                    } else {
if (keywordList.contains(tempdizisi) == true) {
countMatrix[i][keywordList.indexOf(tempdizisi)] =
countMatrix[i][keywordList.indexOf(tempdizisi)] + 1;
                    }
                }
                } // while sonu
            } // while sonu
        }
    }

System.out.println("***** Tum Matris *****");
    //Kelimeler :[edit, countri, abroad., mkrmkr, dewey...
        for (int i = 0; i < 1; i++) {
            System.out.println("Kelimeler:" + keywordList);
        }
        for (int i = 0; i < listOfFiles.length; i++) {
            for (int jj = 0; jj < keywordList.size(); jj++) {
                //sutunları toplama.
countMatrix[listOfFiles.length][jj] += countMatrix[i][jj];
            }
        }
        System.out.println("");
        System.out.print("Toplam:");
    }
    for (int j = 0; j < keywordList.size(); j++) {
        System.out.print("    | ");
        System.out.print(countMatrix[countMatrix.length - 1][j]);
        counttemp[0][j] = countMatrix[countMatrix.length - 1][j];
//countMatrix'inin toplam satırı bu tek boyutlu matrise
kopyalandı
    }
    System.out.println(" -|satırsonu");
    cc = new int[keywordList.size()];
        for (int j = 0; j < keywordList.size(); j++) {

```

```

// ** En önemli bölüm ** toplamda en az 10 defa geçenleri al.
if (counttemp[0][j] >= 10) {
    counter = j;
    cc[tt] = counter;
    tt = tt + 1;
}
}
System.out.println("");
System.out.println("*** Ayıklanmış Matris ***");
for (int kk = 0; kk < tt; kk++) {
    keywordList2.add(keywordList.get(cc[kk]));
}
countMatrix2 = new int[listOfFiles.length][tt];
for (int i = 0; i < listOfFiles.length; i++) {
    for (int yy = 0; yy < tt; yy++) {
        countMatrix[i][yy]=countMatrix[i][cc[yy]];
//matrisin son hali (ayıklanmış)
        countMatrix2[i][yy]=countMatrix[i][cc[yy]];
    }
}
System.out.println("");
for (int i = 0; i < listOfFiles.length; i++) {
    System.out.println(listOfFiles[i].getName());
    for (int yy = 0; yy < tt; yy++) {
        System.out.print("    |    ");
        System.out.print(countMatrix2[i][yy]);
    }
    System.out.println(" -|satırsonu ");
}
System.out.println(""+countMatrix2.length+" : ikinci matrisin
sıra sayısı");
System.out.println(""+keywordList2.size()+" : ikinci matrisin
sütun (ayıklanmış kelime) sayısı");
System.out.println(" ***"+cc.length+" ***");

Cosine cos = new Cosine();
cos.cosineSimilarity();
} // main sonu
} //class sonu

```

#### - Benzerlik oranlarının bulunması

```

import java.util.Vector;
import java.io.File;

public class Cosine extends ReadingMultipleFile {

    public static void cosineSimilarity() {

        double vektorelcarpim = 0.0;    // d1.d2 için
        double uzunluk1 = 0.0;        // ||d1||
        double uzunlukok1 = 0.0;      // ||d1||
        double uzunluk2 = 0.0;        // ||d2||
        double uzunlukok2 = 0.0;      // ||d2||
    }
}

```

```

double cosineSimilarity = 0.0; // d1.d2 / ||d1||||d2||
int f = 0;
int dogru = 0;
double toplam = 0.0;
int yanlis = 0;
double max = 0.0;

Vector<Integer> docVector1 = new
Vector<Integer>(keywordList2.size());
Vector<Integer> docVector2 = new
Vector<Integer>(keywordList2.size());

// cisi için. tüm dokümanların beşte biri eğitim kümesi olsun.
for (int i = 0; i < listOfFiles.length / 5; i++) {
    for (int k = listOfFiles.length / 5; k <
listOfFiles.length; k++) {
        for (int j = 0; j < keywordList2.size(); j++) {

            docVector1.add(countMatrix2[i][j]);
            docVector2.add(countMatrix2[k][j]);

uzunluk1 += docVector1.get(j) * docVector1.get(j); // ||d1||
uzunluk2 += docVector2.get(j) * docVector2.get(j); // ||d2||
vektorelcarpim += docVector1.get(j) * docVector2.get(j);
        }
uzunlukok1 = Math.pow(uzunluk1, 0.5);
uzunlukok2 = Math.pow(uzunluk2, 0.5);
cosineSimilarity = vektorelcarpim / (uzunlukok1 * uzunlukok2);
        if (cosineSimilarity > max) {
            max = cosineSimilarity;
            f = k;
        }
    }
if (k == listOfFiles.length - 1) {
System.out.println(listOfFiles[i].getName() + " dosyasına en
benzer dosya : " +listOfFiles[f].getName() + " Benzerlik oranı
: " + max);
        if (listOfFiles[i].getName().substring(0, 2).equals(
listOfFiles[f].getName().substring(0, 2))) {
            dogru = dogru + 1;
        } else {
            yanlis = yanlis + 1;
File afile = new File
("C:\\Speech_Synthesis_DataSet\\classic3\\"+listOfFiles[i].get
Name());
afile.renameTo(new
File("C:\\Speech_Synthesis_DataSet\\classic3\\yanlis\\"+listOf
Files[i].getName()+"-"+listOfFiles[f].getName()));
        }
    }
    docVector1.removeAllElements();
    docVector2.removeAllElements();
    uzunluk1 = 0.0;
    uzunlukok1 = 0.0;
    uzunluk2 = 0.0;
    uzunlukok2 = 0.0;
    cosineSimilarity = 0.0;
}
}

```



```

        vektorelcarpim = 0.0;
    }
    f = 0;
    max = 0.0;
    uzunluk1 = 0.0;
    uzunlukok1 = 0.0;
    uzunluk2 = 0.0;
    uzunlukok2 = 0.0;
    cosineSimilarity = 0.0;
    vektorelcarpim = 0.0;
}
System.out.println("Dogru sayısı : " + dogru);
System.out.println("Yanlış sayısı : " + yanlis);
    toplam = dogru + yanlis;
System.out.println("Basarim Orani % : " + dogru/toplam * 100);
}
}

```

#### - Ses Dosyalarının elde edilmesi

```

package speech;

import com.sun.speech.freetts.Voice;
import com.sun.speech.freetts.VoiceManager;
// SES DOSYASI (WAV) OLARAK KAYDETMEK İÇİN
import com.sun.speech.freetts.audio.SingleFileAudioPlayer;
import java.io.File;
// TEXT DOSYASINDAN OKUTMAK İÇİN
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

public class Speech {

    static String pathsrc=
        "C:\\Speech_Synthesis_DataSet\\classic3";
    static String pathsrc2 =
        "C:\\Speech_Synthesis_DataSet\\classic3\\yanlis";
    static String dest =
        "C:\\Speech_Synthesis_DataSet\\ses\\";
    static String dest2 =
        "C:\\Speech_Synthesis_DataSet\\yanlis_ses\\";
    public static File destfolder = new File(dest);
    public static File destfolder2 = new File(dest2);
    public static File srcfolder = new File(pathsrc);
    public static File src2folder = new File(pathsrc2);
    public static File[] dlistOfFiles = destfolder.listFiles();
    public static File[] dlistOfFiles2 = destfolder2.listFiles();
    public static File[] slistOfFiles = srcfolder.listFiles();
    public static File[] s2listOfFiles = src2folder.listFiles();

    public static void main(String[] args) throws
    FileNotFoundException, IOException{

```

```

System.setProperty("mbrola.base",
                    "D:\\mali_TEZ_Uygulama\\mbrola");
    VoiceManager voiceManager = VoiceManager.getInstance();
    Voice[] voices = voiceManager.getVoices();
// sistemde tanimli tum ses dosyalarinin adedi
System.out.println("All voices available:" + voices.length);
    for (int i = 0; i < voices.length; i++) {
        System.out.println("    " + voices[i].getName() );
    }
// Ses dosyasi tanimlari -----

    Voice v,v1,v2,v3,v4,v5 ;
    VoiceManager vm=VoiceManager.getInstance();
    v=vm.getVoice("kevin");
    v1=vm.getVoice("kevin16");
    v2=vm.getVoice("alan");
    v3=vm.getVoice("mbrola_us1");
    v4=vm.getVoice("mbrola_us2");
    v5=vm.getVoice("mbrola_us3");
    String cisi ="i";
    String cran ="r";
    String med ="e";
    String wav ="wav";

// Genel Sınıflandırma için
    for (int i = 0; i < slistOfFiles.length-1 ; i++) {
InputStream fs = new
FileInputStream("C:\\Speech_Synthesis_DataSet\\classic3\\"+sli
stOfFiles[i].getName());
SingleFileAudioPlayer sfap = new
SingleFileAudioPlayer("C:\\Speech_Synthesis_DataSet\\ses\\"+sli
stOfFiles[i].getName().substring(0,
10),javax.sound.sampled.AudioFileFormat.Type.WAVE);
if (slistOfFiles[i].getName().substring(1, 2).equals(cisi)) {
// TUM DOSYALARININ WAV FORMATINDA SES DOSYALARI OLUSTU.

System.out.println(i+1+":"+slistOfFiles[i].getName()+" dosyasi
okunuyor.....");
    fs = new
FileInputStream("C:\\Speech_Synthesis_DataSet\\classic3\\"+sli
stOfFiles[i].getName());
        v3.allocate();
        v3.setAudioPlayer(sfap);
        v3.speak(fs);
        sfap.close();

    }else if (slistOfFiles[i].getName().substring(1,
2).equals(cran) ){
// TUM DOSYALARININ WAV FORMATINDA SES DOSYALARI OLUSTU.

System.out.println(i+1+":"+slistOfFiles[i].getName()+" dosyasi
okunuyor.....");
        fs = new
FileInputStream("C:\\Speech_Synthesis_DataSet\\classic3\\"+sli
stOfFiles[i].getName());

```

```

        v1.allocate();
        v1.setAudioPlayer(sfap);
        v1.speak(fs);
        sfap.close();
    }else if (slistOfFiles[i].getName().substring(1,
2).equals("med") ){
// TUM DOSYALARININ WAV FORMATINDA SES DOSYALARI OLUSTU.

System.out.println(i+1+": "+slistOfFiles[i].getName()+" dosyasi
okunuyor.....");
        v4.allocate();
        // Wav dosyasi olarak kaydetmek icin
        v4.setAudioPlayer(sfap);
        v4.speak(fs);
        sfap.close();
        fs.close();
    }
}

// Ses Dosyalarında farklılık oluşturmak istenirse
for (int i = 0; i < s2listOfFiles.length; i++) {
FileInputStream fss = new
FileInputStream("C:\\Speech_Synthesis_DataSet\\classic3\\yanli
s\\"+s2listOfFiles[i].getName());
SingleFileAudioPlayer sfap = new
SingleFileAudioPlayer("C:\\Speech_Synthesis_DataSet\\yanlisSes
\\"+s2listOfFiles[i].getName(),javax.sound.sampled.AudioFileFo
rmat.Type.WAVE);

if (s2listOfFiles[i].getName().substring(17,18).equals("cisi")){
System.out.println(" Sadece CISI olarak siniflandirilanlar");
v3.allocate();
v3.setDurationStretch(1.0f); //0.5f cok hizli, 2.0f cok yavas
v3.setPitchRange(11.0f);// daha dogal 50.0f
v3.setPitch(20); // ince:200 kalin:20
v3.setVolume(0.78f);
v3.setAudioPlayer(sfap);
v3.speak(fss);
sfap.close();
fss.close();

}else if (s2listOfFiles[i].getName().substring(17,
18).equals("cran") ){
System.out.println(" Sadece CRAN olarak siniflandirilanlar");
        v1.allocate();
        v1.setDurationStretch(1.0f);
        v1.setPitchRange(11.0f);
        v1.setPitch(20);
        v1.setVolume(0.78f);
        v1.setAudioPlayer(sfap);
        v1.speak(fss);
        sfap.close();
        fss.close();

}else if (s2listOfFiles[i].getName().substring(17,
18).equals("med") ){

```

```
System.out.println(" Sadece MED olarak siniflandirilanlar.");
    v4.allocate();
    v4.setDurationStretch(1.0f);
    v4.setPitchRange(11.0f);
    v4.setPitch(20);
    v4.setVolume(0.78f);
    v4.setAudioPlayer(sfap);
    v4.speak(fss);
    sfap.close();
    fss.close();
    }
    }
System.out.println(" ----- The End -----");
}
}
```



## ÖZGEÇMİŞ



### YAYINLAR

Şirin Y. ve Kutlugün M.A. (2017). *Konuşma Sentezinde Artan Doğallık için Boyut Azaltma Seçimi*. 25th Signal Processing and Communications Applications Conference (SIU 2017), Antalya, (IEEE : s. 10.1109/SIU.2017.7960573).

Kutlugün M.A. ve ark. (2017). *Yapay Sinir Ağları ve K-En Yakın Komşu Algoritmalarının Birlikte Çalışma Tekniği (Ensemble) ile Metin Türü Tanıma*. İstanbul: XXII. Türkiye’de İnternet Konferansı (inet’tr17), Bahçeşehir Üniversitesi.

Kutlugün M.A. ve ark. (2017). *Derin Sinirsel Ağlarla Konuşma Tespiti ve Cinsiyet Tahmini*, İstanbul: XXII. Türkiye’de İnternet Konferansı (inet’tr17), Bahçeşehir Üniversitesi.