

**T.C.
TURGUT ÖZAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**TÜRKÇE EĞİTSEL İÇERİKLİ VERİLERİN
OKUNABİLİRLİK SEVİYELERİNE GÖRE
SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Mustafa Anıl TÖRER**

**Tez Danışmanı
Yrd. Doç. Dr. Rifat ÖZCAN**

Ankara-2015

**T.C.
TURGUT ÖZAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**TÜRKÇE EĞİTSEL İÇERİKLİ VERİLERİN OKUNABİLİRLİK
SEVİYELERİNE GÖRE SINIFLANDIRILMASI**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Mustafa Anıl TÖRER**

**Tez Danışmanı
Yrd. Doç. Dr. Rıfat ÖZCAN**

Ankara-2015

Bilimsel Etik Bildirim Sayfası

Turgut Özal Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,

- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,

- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,

- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,

- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,

- Bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı beyan ederim.

11/09/2015

Mustafa Anıl TÖRER

ONAY

Mustafa Anıl TÖRER tarafından hazırlanan “*Türkçe Eğitsel İçerikli Verilerin Okunabilirlik Seviyelerine Göre Sınıflandırılması*” başlıklı bu çalışma, 09.09.2015 tarihinde yapılan savunma sınavı sonucunda *oybirliği* ile başarılı bulunarak jürimiz tarafından *Elektrik ve Bilgisayar Mühendisliği Anabilim* dalında *Yüksek Lisans* tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Rıfat ÖZCAN

Yrd. Doç. Dr. Muhammet BAŞTAN

Yrd. Doç. Dr. İsmail Sengör ALTINGÖVDE

ÖNSÖZ

Bu tez çalışmasında Milli Eğitim Bakanlığı'nın yayınladığı Türkçe ders kitapları analiz edilerek oluşturulan sınıflandırma modeline göre internet ortamında bulunan eğitsel içerikli verilerin sınıflandırılması amaçlanmış ve Turgut Özal Üniversitesi Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tez çalışması olarak gerçekleştirilmiştir. Bu çalışma TÜBİTAK 113E065 numaralı projesi (Arama Teknolojilerinin İlk ve Orta Öğretim Öğrencilerine Göre Optimizasyonu) kapsamında desteklenmiştir.

Tez çalışmamız boyunca verdiği motivasyon ve tüm gayretlerinden dolayı kıymetli danışmanım Yrd. Doç. Dr. Rıfat ÖZCAN'a teşekkürü bir borç bilirim.

En az danışmanım kadar desteklerini esirgemeyen, çalışmalarımızı yönlendiren değerli hocam Yrd. Doç Dr. Muhammet BAŞTAN' a teşekkürlerimi sunarım.

Çalışmamızın sonunda, yardımlarını esirgemeyen pek değerli hocam Yrd. Doç. Dr. İsmail Sengör ALTINGÖVDE' ye şükranlarımı sunarım.

Benim her an yanımda olan, tez çalışmaları boyunca gayretlerime destek olan, verdiği moralle çalışmalarımı hızlandırmamı sağlayan çok değerli eşime sevgilerimi sunarım.

ÖZET

TÖRER, Mustafa Anıl. Türkçe Eğitsel İçerikli Verilerin Okunabilirlik Seviyelerine Göre Sınıflandırılması, Yüksek Lisans Tezi, Ankara, 2015.

Bu tez çalışmasında, eğitim çağındaki çocukların okunabilirlik seviyesindeki verilere daha kolay ulaşabilmesini sağlamak amacıyla Türkçe dili için daha başarılı okunabilirlik seviye tespiti yapılması hedeflenmiştir. Çalışmada, Türkçe için geliştirilen Ateşman ve Çetinkaya-Uzun okunabilirlik formülleri, farklı dillerde kullanılmış otomatik okunabilirlik dizini (ARI) formülü ve tespit edilen bazı yeni öznitelikler kullanılmıştır. Milli Eğitim Bakanlığınca yayınlanan 1, 2, 3, 4, 5, 6, 7 ve 8. sınıf Türkçe ders kitapları ile 9, 10, 11 ve 12. sınıf Türk Dili ve Edebiyatı ders kitapları, bu tez çalışmasında analiz edilmiştir. Bu kitaplardan sınıflandırma algoritmaları ile modeller oluşturulmuş ve 14 farklı eğitsel içerikli web sitesinden elde edilen veriler üzerinde test edilmiştir.

Tezin sonucunda, önerilen kelime öznitelikleri okunabilirlik formülleri ile birlikte kullanıldığında, okunabilirlik formüllerine göre daha iyi sonuçlar elde ettiği gözlemlenmiştir.

Anahtar Sözcükler

1. Okunabilirlik
2. Okunabilirlik Formülleri
3. Eğitsel İçerikli Verilerin Sınıflandırılması

ABSTRACT

TÖRER, Mustafa Anl. Classifying Readability Level of Educational Content in Turkish Language, Master Thesis, Ankara, 2015.

In this thesis, it is aimed to detect the level of readability for the Turkish language more successfully, in order to provide easier access to the data in readability level of children of school age. Ateşman and Çetinkaya-Uzun readability formulas developed for Turkish, Automated Readability Index (ARI) formula that is used in different languages and some new features that are detected were used in this study. 1st, 2nd, 3rd, 4th, 5th, 6th, 7th and 8th-Grade Turkish Language course textbooks and 9th, 10th, 11th and 12th-Grade Turkish Language and Literature course textbooks published by Turkish Ministry of National Education were analyzed. From these books, models were created with classification algorithms and it has been tested on educational data obtained from 14 different web sites.

As a result of the thesis, the readability formulas with the suggested word features achieved more successful readability level detection than the readability formulas without them.

Keywords

1. Readability
2. Readability Formulas
3. Classification Of Educational Data

İÇİNDEKİLER

ÖNSÖZ	i
ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER	iv
TABLolar DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
GİRİŞ	1

BİRİNCİ BÖLÜM

OKUNABİLİRLİK KAVRAMI VE LİTERATÜR ÖZETİ

1.1. OKUNABİLİRLİK KAVRAMI	4
1.2. OKUNABİLİRLİK FORMÜLLERİ	5
1.2.1. Dale-Chall Okunabilirlik Formülü	5
1.2.2. Flesch Okunabilirlik Formülü	6
1.2.3. Gunning Sis Dizini	9
1.2.4. Otomatik Okunabilirlik Dizini (ARI)	9
1.2.5. Fry Okunabilirlik Grafiği	10
1.2.6. Ateşman Okunabilirlik Formülü	10
1.2.7. Çetinkaya-Uzun Okunabilirlik Formülü.....	11
1.3. YABANCI DİLLER İÇİN GELİŞTİRİLEN FORMÜLLERİN KULLANILMASI.....	12
1.4. İLGİLİ ÇALIŞMALAR	13

İKİNCİ BÖLÜM

EĞİTSEL İÇERİLİKLİ VERİLERİN SINIFLANDIRILMASI

2.1. VERİ DÖNÜŞÜM VE HAZIRLAMA İŞLEMLERİ	17
2.2. KONU BELİRLEME İŞLEMLERİ	17
2.3. CÜMLE TESPİT İŞLEMLERİ	18

2.4. KELİME-HECE İŞLEMLERİ	18
2.5. ÖZİNİTELİK İŞLEMLERİ	18
2.5.1. Kelime Listesi Öznitelikleri	19
2.5.2. Hece Öznitelikleri	19
2.5.3. Oran Öznitelikleri	20
2.6. SINIFLANDIRMA İŞLEMLERİ	20
2.7. METİN ANALİZ PROGRAMI VE KULLANILMASI	20
2.7.1. Veri Giriş-Çıkışı ve Konuların Belirlenmesi	22
2.7.2. Metin Analiz Programının Kullanılması	22
2.7.2.1. Yeni Analiz Fonksiyonu	23
2.7.2.2. Veri Sorgulama Fonksiyonu	26
2.7.2.3. Veri Al / Ver Fonksiyonu	28
2.7.2.4. Normalizasyon Fonksiyonu	28
2.8. WEKA PROGRAMI	29

ÜÇÜNCÜ BÖLÜM

DENEYSEL SONUÇLAR

3.1. VERİLER	31
3.1.1. Veri Ön Hazırlık	31
3.2. DENEYSEL SONUÇLAR	32
3.3. KADEME BAZLI SONUÇLAR	32
3.3.1. Ateşman, Çetinkaya-Uzun ve ARI Okunabilirlik Formülleri Sonuçları	32
3.3.2. Tekli Öznitelik Grubu Sonuçları	33
3.3.3. İkili Öznitelik Grubu Sonuçları	34
3.3.4. Üçlü Öznitelik Grubu Sonuçları	35
3.3.5. Dörtlü Öznitelik Grubu Sonuçları	35
3.3.6. Kademe Bazlı Sonuçların Değerlendirilmesi	36
3.4. SINIF BAZLI SONUÇLAR	37

3.5. TESPİT EDİLEN EN İYİ ÖZNİTELİKLER İLE VARSAYIMLI	
SONUÇLAR	37
SONUÇ	39
KAYNAKÇA	41
EKLER	44



TABLolar DİZİNİ

Tablo 1. Dale-Chall okunabilirlik formülü	6
Tablo 2. Dale-Chall okunabilirlik seviyeleri	7
Tablo 3. Flesch okunabilirlik formülü	7
Tablo 4. Flesch-Kincaid sınıf seviyesi formülü	8
Tablo 5. Flesch-Kincaid tahmini sınıf seviyeleri	8
Tablo 6. Gunning sis dizini formülü	9
Tablo 7. Otomatik okunabilirlik dizini formülü	10
Tablo 8. Ateşman okunabilirlik formülü	10
Tablo 9. Ateşman okunabilirlik seviyeleri	11
Tablo 10. Çetinkaya-Uzun okunabilirlik formülü	11
Tablo 11. Çetinkaya-Uzun okunabilirlik seviyeleri	12
Tablo 12. Örnek cümle ve okunabilirlik seviyeleri	13
Tablo 13. Weka programı ile kullanılan 22 farklı algoritma listesi	30

ŞEKİLLER DİZİNİ

Şekil 1. Eğitsel içerikli verilerin analizindeki süreçler	16
Şekil 2. Metin analiz programı iş akış diyagramı.....	21
Şekil 3. Metin analiz programı giriş ekranı.....	21
Şekil 4. Metin analiz programı konu ve alt konu ayraçları.....	22
Şekil 5. Metin analiz programı yeni analiz ekranı	23
Şekil 6. Metin analiz programı çıktı ayarları	24
Şekil 7. Metin analiz programı sonuçların hesaplanması	25
Şekil 8. Metin analiz programı sonuçlar sekmesi	27
Şekil 9. Metin analiz programı veri tabanı sorgulama ekran	27
Şekil 10. Metin analiz programı veri al / ver ekranı	28
Şekil 11. Metin analiz programı normalizasyon ekranı	29
Şekil 12. Weka programı iş akış diyagramı	30
Şekil 13. Kademe bazında okunabilirlik formülleri sonuç grafiği	33
Şekil 14. Kademe bazında tekli öznitelik grupları sonuç grafiği	34
Şekil 15. Kademe bazında ikili öznitelik grupları sonuç grafiği	34
Şekil 16. Kademe bazında üçlü öznitelik grupları sonuç grafiği	35
Şekil 17. Kademe bazında dördü öznitelik grubu sonuç grafiği	36
Şekil 18. Kademe bazında en iyi olarak belirlenen öznitelik grupları sonuç grafiği	36
Şekil 19. Sınıf bazında en iyi beş öznitelik grubu sonuç grafiği	37

GİRİŞ

Tüm dünyada teknolojinin hızla gelişmesiyle dijital veriler (web sayfaları) hızlı bir biçimde büyümektedir. Gelişen teknoloji karşısında hemen hemen tüm sektörlerde bilgisayar, kullanılan en önemli araçlardan birisi haline gelmiştir. Bilgisayar ve elektronik alandaki gelişmeler sayesinde, yaptığımız işlemlerin çoğunda, bu araçların kullanılarak gerçekleştirilmeye başlanmış ve yapılan tüm işlemler dijital ortamlarda saklanmaya başlanmıştır. Zaman geçtikçe biriken veriler inanılmaz boyutlara ulaşmakta ve bu verilerden anlamlı bilgiler çıkarabilmek gün geçtikçe daha da zor hale gelmektedir.

Teknolojinin gelişmesiyle ülkemizdeki bilgisayar ve internet kullanım oranları da artmaktadır. Günümüzde insanlar istedikleri veriye dijital ortamlardan erişebilmektedir. TÜİK'in yaptığı son araştırmalara göre 2004 ile 2015 yılları arasında bilgisayar ve internet kullanım sayılarında belirgin seviyelerde artış gözlemlenmiştir. 16-74 yaş arası bireylerde yapılan araştırmaya göre; 2004 yılında bilgisayar kullanım oranı %23.6 iken 2015 yılında %54.8'e, internet kullanım oranı 2004 yılında %18.8 iken 2015 yılında %55.9'a yükselmiştir. En çok değişim, hanelerde internet erişiminde görülmüştür. Bu oran 2004 yılında %7 iken 2015 yılında %69.5'e yükselmiştir (TÜİK, 2015).

Arama motorları, dijital ortamlardaki verilerin kullanıcıya ulaştırılması amacıyla geliştirilen sistemlerdir. Gelişmiş düzeydeki arama motorları, yazılımsal ve donanımsal alt yapıya sahip sistemler olup, temel olarak internet ortamındaki verileri kendi algoritmalarınca endeksleme (index) işleminden geçirir. Endekslenen verilerden kullanıcıların yaptığı sorgulara göre, aramalara ilgili sonuçları hızlı bir şekilde verebilmesi hedeflenmektedir. Günümüzde bilgi erişiminin kolaylaştırılmasında arama motorlarının etkisi büyüktür. Arama motorları ile dünyadaki dijital verilere internet üzerinden çevrimiçi erişilebilir. Ancak arama motorları kişiye göre optimize edilmemiş, genel amaçlı kullanım olduğu için genellikle yetişkin kullanıcılara göre tasarlanmıştır. Ancak internet erişimindeki

kolaylıklar ve artan akıllı telefon kullanımı ile çocuklar da arama motorlarını sıkça kullanmaktadırlar. Bilal (2013), yaptığı çalışmasında çocukların okuma ve yaş seviyelerine uygun arama yapabilecekleri özel arama motorlarının bulunduğunu ancak arama motorlarının çoğunluğunun çocuklar yerine genel kullanım amaçlı kullanılacak şekilde optimize edildiğini belirtmiştir. Öğrencilerin artan dijital eğitsel içeriklere erişimi de önemli bir problemdir. Öğrencilerin okuma düzeyleri yetişkinlerden farklı olduğu gibi sınıf seviyelerine göre de farklılık göstermektedir. Bu nedenle öğrencilerin okuma seviyelerine uygun eğitsel içeriklere erişmesini sağlamak önemli bir problemdir.

Eğitim gören ilkökul, ortaokul veya lise seviyesindeki öğrencilerin sınıflara veya kademelere (ilkokul, ortaokul, lise) göre arama motorları ile aradıkları eğitsel içerikli dijital verilerin (web sayfalarının), arayan kişinin okunabilirlik seviyesine uygun verilere daha hızlı ve daha kolay erişebilmesi bu tez çerçevesinde önem arz etmektedir. Bu bağlamda şu sorulara cevap aranmıştır:

- Literatürde önerilen okunabilirlik formüllerinin K-12 seviyesinde Türkçe eğitsel içeriklerinin okunabilirliklerini sınıflandırmadaki başarıları ne düzeydedir?
- Eğitsel içeriklerin sınıflandırılmasında başarıyı artıracak yeni/özgün öznitelikler bulunabilir mi? Özniteliklerin sınıflandırma performansına etkileri nelerdir?
- Hangi sınıflandırma algoritması ile en iyi başarı elde edilebilmektedir?

Bu tez çalışması 3 bölümden oluşmaktadır:

1. bölümde okunabilirlik kavramı, tarihsel gelişimi, okunabilirlik formülleri hakkında bilgiler ile literatür taraması yapılarak bu tez çalışması ile ilgili çalışmaların özeti sunulmaktadır.

2. bölümde verilerden hazırlanan öznitelikler ve uygulanan sınıflandırma teknikleri hakkında bilgilendirme yapılmaktadır.

3. bölümde veriler ve veri hazırlık işlemlerinden bahsedilerek elde edilen deneysel sonuçların analizleri hakkında bilgiler verilmektedir.

Sonuç kısmında elde edilen bulgular ile ilgili yorumlar ve ileride yapılması gereken çalışmalardan bahsedilmiştir.



BİRİNCİ BÖLÜM

OKUNABİLİRLİK KAVRAMI VE LİTERATÜR ÖZETİ

1.1. OKUNABİLİRLİK KAVRAMI

İlkokul yıllarından itibaren öğrenme işlevini yerine getirdiğimiz en önemli araçlardan bir tanesi okumaktır. İnsanlar yaşamları boyunca kişisel gelişim ve bilgi edinim için araştırma ve öğrenme süreçlerinden geçmektedir. Bu süreçlerin temel şartı okuma ve okuduğunu anlayabilmektir. Okunan metnin anlaşılabilmesi, kişilerin yaptığı işlevden zevk alması demektir.

Ateşman'a (1997) göre okunabilirlik kavramı, metinlerin okuyan tarafından ne kadar kolay ya da güç anlaşılır olduğudur. Metnin okunabilirlik seviyesinin yüksek olması anlaşılmayı zorlaştırırken, metnin kolay olması da okumaya olan ilgiyi de azaltmaktadır (Ateşman, 1997; Temur, 2003). Güneş (2000), yaptığı çalışmasında ilgi çeken metinlerin, diğerlerine göre daha hızlı ve kolay okunduğu sonucuna ulaşmıştır. Richards'a göre okunabilirlik, yazılı materyallerin nasıl kolayca okunabilir ve anlaşılabilir olduğudur. Okunabilirlik metindeki ortalama cümle uzunluğu, metnin içerdiği yeni kelimelerin sayısı ve kullanılan dildeki gramatik karmaşıklığa göre değişir (Richards ve diğ, 1992). Dale ve Chall'a (1948) göre okunabilirlik, basılı materyal ile ilgili bütün unsurların toplamı, başarı ise onu anlama, uygun hızda okuma ve ilginç bulmadır. Klare'e (1963) göre okunabilirlik, yazım şeklinden ötürü anlama ve kavrama kolaylığıdır. Klare okunabilirlik kavramını okunaklık, anlaşılabilirlik ve okunabilirlik başlıkları altında üç anlamda kullanmaktadır. Dubay'e (2004) göre okunabilirlik, bazı metinleri diğerlerine göre kolay okunabilir yapmadır. Genellikle yazı tipi ve düzeni ile ilgilenen okunaklık (legibility) kavramı ile karışmaktadır.

Tüm bu tanımlara göre okunabilirlik, okuyan kişinin okuduğu metni ne kadar anladığının bir göstergesidir. Yani kaynak ile alıcı arasındaki anlaşılma uyumudur denilebilir.

1.2. OKUNABİLİRLİK FORMÜLLERİ

Metinlerin okuyucuya uygunluğu 19. yy.'da sorgulanmaya başlansa da Dubay (2004), okunabilirlik formülleri ile ilgili çalışmalarda 1921 yılında E.L. Thorndike'in yaptığı "Öğretmen kelime kitabı" çalışmasını dayanak göstermektedir. Bu kitap 41 farklı kaynağın taranarak İngilizcede sık kullanılan 10,000 kelimedenden oluşmuştur (Thorndike, 1921).

Okunabilirlik üzerine yapılan çalışmalarda kişilerin okudukları metnin ilgi alanları veya farklı alanlarda olsa bile seviyelerine uygun metinlerin bulunabilmesi hedeflenmektedir.

Chall (1996), yaptığı araştırmada ellinin üzerinde okunabilirlik formülünün olduğunu ancak bunların beşinin yaygın olarak kullanıldığını, bunların da Lorge (1944), Flesch (1948), Dale & Chall (1948), Bormuth (1969), ve Fry'ın (1977) okunabilirlik formülleri olduğunu belirtmiştir.

Okunabilirlik ile ilgili çalışmalar 1950'lere kadar pek fazla yaygınlaşmamasına rağmen Rudolf Flesch, George Klare, Edgar Dale, and Jeanne Chall gibi yazarların geliştirdiği formüller ve yaptığı araştırmalar sayesinde okunabilirlik formülleri yaygın olarak gazetecilik, araştırma, sağlık, hukuk, sigorta ve sanayide kullanılmaya başlanmıştır. 1980'li yıllarda ise okunabilirlik kavramı ile ilişkili 1000'in üzerinde çalışma yayınlanmış, teorik ve istatistiksel geçerliği doğrulanmış 200 okunabilirlik formülü bulunmaktadır (Dubay, 2004).

Bu tez kapsamında yabancı kelimelere yönelik üretilen en meşhur formüllerden 5 tanesinden, Türkçe üzerine geliştirilen formüllerin 2 tanesinden bahsedilmiştir.

1.2.1. Dale-Chall Okunabilirlik Formülü

Dale-Chall okunabilirlik düzeyini tespit edebilmek için kelime ve cümle sayıları ile kendilerinin belirledikleri, basit ve anlaşılır kabul edilen kelime listelerini kullanarak 1948 yılında "Okunabilirliği kestirmek için bir formül (A formula for

predicting readability)" adlı çalışmasında bu formülü geliştirmişlerdir (Dale ve Chall, 1948).

İlk oluşturulan kelime listesi 769 tane olmasına rağmen daha sonra bu liste 1995 yılında güncellenerek 3000 kelimededen oluşan, 4. sınıf öğrencilerinin %80'inin anlayabileceği yeni kelime listesi kullanılmaya başlanmıştır.

Bu formülün farkı diğer okunabilirlik formüllerinin aksine kelime listesinin kullanılmasıdır. Dale-Chall okunabilirlik formülüne göre bilindik kelimelerin kullanılması ile metin, daha kolay okunabilir hale gelmektedir.

Formüle göre seçilen metnin kelime sayısı olarak ortalama cümle uzunluğu ve metindeki kelimelerin ne kadarının hazırlanan 3000 kelimelik listenin dışında kaldığının belirlenmesi ile hesaplama yapılmaktadır.

Tablo 1'de gösterilen formüle göre çıkan sonuçlar Tablo 2'deki karşılaştırmaya göre sınıf seviyesi belirlenmektedir.

Tablo 1. Dale-Chall okunabilirlik formülü

$DCOP = (0.1579 \times KO) + (0.0496 \times OCU) + 3.6365$	
DCOP (Dale-Chall Okunabilirlik Puanı)	Formül sonucunda çıkan okunabilirlik puanı
KO (Zor Kelime Oranı)	Dale-Chall'ın bilindik 3000 kelimelerinde bulunmayan kelimelerin yüzdesi
OCU(Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü

1.2.2. Flesch Okunabilirlik Formülü

1948 yılında Flesch tarafından geliştirilen, Tablo 3'te verilen formül kelime olarak cümle uzunluğu ve hece olarak kelime uzunlukları arasındaki ilişkiye göre sonuç vermektedir (Flesch, 1948).

Tablo 2. Dale-Chall okunabilirlik seviyeleri

Dale-Chall Okunabilirlik Puanı	Okunabilirlik Seviyesi
4.9 ve aşağısı	4. sınıf ve alt sınıflar
5.0 - 5.9	5 - 6. sınıf
6.0 - 6.9	7 - 8. sınıf
7.0 - 7.9	9 - 10. sınıf
8.0 - 8.9	11 - 12. sınıf
9.0 - 9.9	13 - 15. sınıf (Kolej)
10+	16. sınıf ve üzeri (Kolej Mezunu)

Tablo 3. Flesch okunabilirlik formülü

$FKOP = 206.835 - (1.015 \times OCU) - (84.6 \times OKU)$	
FKOP (Flesch Kolay Okuma Okunabilirlik Puanı)	Formül sonucunda çıkan okunabilirlik puanı
OCU (Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü
OKU (Ortalama Kelime Uzunluğu)	Hece sayısının kelime sayısına bölümü

İncelenen metindeki cümle sayıları, kelime sayıları ve bu kelimelerin hece sayılarının belirlenmesi sonucunda kelime sayısının cümle sayısına bölünmesi ile ortalama cümle uzunluğu (OCU), hece sayısının kelime sayısına bölünmesi ile de ortalama kelime uzunluğu (OKU) belirlenmiş olur. Bulunan değerler formülde yerlerine konularak okunabilirlik puanı elde edilmektedir.

Tablo 4. Flesch-Kincaid sınıf seviyesi formülü

$FKSS = (0.39 \times OCU) + (11.8 \times OKU) - 15.59$	
FKSS (Flesch-Kincaid Sınıf Seviyesi)	Formül sonucunda çıkan sınıf seviyesi
OCU (Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü
OKU (Ortalama Kelime Uzunluğu)	Hece sayısının kelime sayısına bölümü

1975 yılında bu formüle göre çıkan sonuçların sınıf seviyesine göre karşılığının bulunması için Kincaid çalışmalar yapmış ve Tablo 4’te verilen Flesch-Kincaid sınıf seviyesi formülünü geliştirmiştir (Kincaid ve diğ, 1975).

Formül sonucunda Amerikan sınıf seviyelerine göre seviye tespiti yapılmaktadır. Tablo 5’de Flesch kolay okuma formülüne göre okunabilirlik seviyesi ve Flesch-Kincaid sınıf seviyesi sonucu oluşan tahmini sınıf seviyesi gösterilmiştir.

Tablo 5. Flesch-Kincaid tahmini sınıf seviyeleri

Flesch Okunabilirlik Puanı	Okunabilirlik Seviyesi	Tahmini Sınıf Seviyesi
90 - 100	Çok Kolay	4. sınıf
80 - 90	Kolay	5. sınıf
70 - 80	Oldukça Kolay	6. sınıf
60 - 70	Standart	7 - 8. sınıf
50 - 60	Oldukça Zor	Lise
30 - 50	Zor	Kolej
0 - 30	Çok Zor	Kolej Mezunu

1.2.3. Gunning Sis Dizini

Amerikalı bir iş adamı olan Robert Gunning 1952 yılında İngilizce metinlerin okunabilirlik seviyelerinin tespiti amacıyla Gunning sis dizini (Gunning fog index) formülünü geliştirmiştir (Gunning, 1952).

Tablo 6’ da verilen formül kelime sayısının cümle sayısına bölünmesi ile üç ve daha fazla heceli kelime sayısının geriye kalan kelime sayısına bölünmesinden oluşmaktadır.

Tablo 6. Gunning sis dizini formülü

GSDP = 0.4 x [OCU + (100 x KKU)]	
GSDP (Gunning Sis Dizini Puanı)	Formül sonucunda çıkan sis dizini puanı
OCU(Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü
KKU (Karışık Kelime Uzunluğu)	Üç ve daha fazla heceli kelime sayısının geriye kalan kelime sayısına bölümü

Burada dikkat edilmesi gereken konu üç ve daha fazla heceli kelimelerin hesaplanmasıdır. Bu hesaplamayı yaparken özel isimler, bilinen jargonlar ve birleşik kelimeler ile İngilizcede sık kullanılan “-es, -ed, -ing” gibi ekler hece sayısına dahil edilmemektedir. Formül sonucunda çıkan sonuç, eğer 8-10 arasında ise kolay, 11’den yukarı ise zor bir metin olduğunu göstermektedir (Stone, 1996; Güneş, 2000:340; Goldbort, 2001:40; Temur, 2003).

1.2.4. Otomatik Okunabilirlik Dizini (ARI)

Senter ve Smith tarafından 1967 yılında geliştirilen bu formül (Tablo 7), diğer formüllerin aksine metindeki ortalama sözcük uzunluğunun kaç harften oluştuğu niceliğini de göz önünde bulundurmıştır (Senter ve Smith, 1967). Formülün geliştirilmesindeki asıl amaç elektrikli daktiloların gerçek zamanlı izlenebilmesidir.

Tablo 7. Otomatik okunabilirlik dizini formülü

$ARI = (4.71 \times OKU) + (0.6 \times OCU) - 21.43$	
ARI (Otomatik okunabilirlik endeksi)	Formül sonucunda çıkan okunabilirlik puanı
OKU (Ortalama Kelime Uzunluğu)	Harf sayısının kelime sayısına bölümü
OCU(Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü

1.2.5. Fry Okunabilirlik Grafiği

İngilizce metinler için 1968 yılında Fry tarafından geliştirilmiş okunabilirlik grafiğidir. Metnin başından, ortasından ve sonundan alınan 100 kelimelik gruplarda ortalama hece sayısı ve ortalama cümle sayıları hesaplanarak üç metinden çıkan sonuçların ortalaması alınır. Çıkan sonuçlar grafikte yerlerine koyularak okunabilirlik seviyesi tespit edilir (Fry, 1968).

1.2.6. Ateşman Okunabilirlik Formülü

1997 yılında Ateşman tarafından Türkçe için geliştirilen ilk formüldür. Ateşman, dil dergisinde yayınlanan “Türkçede okunabilirliğin ölçülmesi” adlı çalışmasında Flesch okunabilirlik formülünü temel alarak Türkçe’ye uyarlamıştır. Formülünde hece olarak ortalama kelime uzunluğu ve kelime olarak ortalama cümle uzunluğu değişkenleri kullanılarak, bunların Türkçe için katsayılarının tespiti yapılmıştır (Ateşman, 1997). Tablo 8’ de geliştirilen okunabilirlik formülü verilmiştir.

Tablo 8. Ateşman okunabilirlik formülü

$AOP = 198.825 - (40.175 \times OKU) - (2.610 \times OCU)$	
AOP (Ateşman Okunabilirlik Puanı)	Formül sonucunda çıkan okunabilirlik puanı
OKU (Ortalama Kelime Uzunluğu)	Hece sayısının kelime sayısına bölümü
OCU(Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü

Formüle göre çıkan sonuç 0-100 aralığındadır. Çıkan sonuca göre Tablo 9’da gösterilen okunabilirlik seviyesi belirlenmektedir.

Tablo 9. Ateşman okunabilirlik seviyeleri

Ateşman Okunabilirlik Puanı	Okunabilirlik Seviyesi
90 - 100	Çok Kolay
70 - 89	Kolay
50 - 69	Orta Güçlükte
30 - 49	Zor
1 - 29	Çok Zor

1.2.7. Çetinkaya-Uzun Okunabilirlik Formülü

Türkçe dili üzerinde yapılan çalışmalarda geliştirilen bir diğer formül, “Türkçe metinlerin düzeylerinin tanımlanması ve sınıflandırılması” adlı doktora tezi ile 2010 yılında Çetinkaya tarafından geliştirilen formüldür. Türkçedeki metinlerin niceliksel özelliklerini ele alarak çeşitli istatistiksel analizler yapılmış ve Tablo 10’daki formül geliştirilmiştir (Çetinkaya, 2010).

Tablo 10. Çetinkaya-Uzun okunabilirlik formülü

$\text{ÇUOP} = 118.823 - (25.987 \times \text{OKU}) - (0.971 \times \text{OCU})$	
ÇUOP (Çetinkaya-Uzun Okunabilirlik Puanı)	Formül sonucunda çıkan okunabilirlik puanı
OKU (Ortalama Kelime Uzunluğu)	Hece sayısının kelime sayısına bölümü
OCU(Ortalama Cümle Uzunluğu)	Kelime sayısının cümle sayısına bölümü

Bu formül sonucunda elde edilen puan Tablo 11’de belirtilen seviyelere göre kategorilendirilmiştir. Tablo 11’de bulunan okunabilirlik seviyelerinden bağımsız

okuma seviyesi, okuyanın herhangi bir yardıma ihtiyaç duymadan metni anlamlandırabileceği seviyedir. Eğitsel okuma seviyesi, okuyanın bilişsel seviyesinin üstünde, eğitici yardımıyla metni anlamlandırabileceği seviyedir. Engelli okuma düzeyi ise okuyanın bilişsel seviyesinin üstünde, eğitimci yardımı olmasına rağmen metni anlamlandıramayacağı seviye olarak belirtilmiştir (Çetinkaya, 2010).

Tablo 11. Çetinkaya-Uzun okunabilirlik seviyeleri

Çetinkaya-Uzun Okunabilirlik Puanı	Okunabilirlik Seviyesi	Eğitim Seviyesi
0 - 34	Engelli Okuma	10, 11, 12. sınıf
35 - 50	Eğitsel Okuma	8, 9. sınıf
51+	Bağımsız Okuma	5, 6, 7. sınıf

1.3. YABANCI DİLLER İÇİN GELİŞTİRİLEN FORMÜLLERİN KULLANILMASI

Okunabilirlik formüllerinin büyük bir kısmı İngilizce dili için tasarlanmıştır. Sönmez'in (2003) çalışmasında Gunning sis dizininin tutarlı bir sonuç vermediği tespit edilmiştir. Köse'nin (2009) yaptığı çalışmasında "Gunning sis dizini" ve "Flesch okuma kolaylığı" formüllerinin Türkçe metinlerde yapılan analizlerinde, okunabilirlik seviyelerinin tespitinde tutarsız sonuçlar verdiği açıkça ortaya konulmaktadır. Bunlara rağmen Marije de Heus ve Djoerd Hiemstra "Webte okunabilirlik: 1 milyar web sayfası ile çalışma (Readability of the web: A study on 1 billion web pages)" adlı araştırmasında farklı dillerden web siteleri analiz edilmiş ve bu analizde ARI'nin kullanıldığını belirtmişlerdir. Smith ve Senter'in ARI'yi özellikle İngilizce dili için tasarlamadığını, buna rağmen sadece İngilizce dilinde test ettiklerini belirterek, literatür çalışmalarında ARI okunabilirlik formülünün farklı diller üzerinde seviye belirleme tespiti yapıp yapmadığı ile ilgili bir çalışma bulamadığını ifade etmişlerdir (Heus ve Hiemstra, 2013).

1.4. İLGİLİ ÇALIŞMALAR

Ülkemizde ilk çalışma Ateşman'ın 1997 yılında yaptığı “Türkçede okunabilirliğin ölçülmesi” çalışmasıdır. Çiftçi ve diğ. (2007), yaptığı araştırmalarında 2006-2007 eğitim öğretim yılında ilköğretim altıncı sınıfta okutulan ders kitaplarından öyküleyici ve bilgilendirici 46 metnin Ateşman formülü ile değerlendirmiş ve kitaplar arasında okunabilirlik açısından bir uyum olmadığı gibi her kitabın, kendi içinde de okunabilirlik seviyesi bakımından da tutarlı olmadığını belirtmiştir. Çoban'ın (2014) yaptığı derleme çalışmasında, Türkçe için yapılan çalışmaların 13'ü dil eğitimi, 6'sı disiplinler arası olmak üzere 19 makale, 5'i yüksek lisans 1'i doktora olmak üzere toplamda 6 tez çalışmasının yapıldığını belirtmiş ve bu çalışmalarda genel olarak mevcut formüllerin Türkçe ders kitaplarına uygulandığını ifade etmiştir.

Okunabilirlik formülleri metnin güçlüğü ile ilgili sadece fikir verdiği yapılan çalışmalarda görülmektedir. Okunabilirlik ölçüsü, metnin nitel özellikleri ile birlikte ele alınırsa ancak kesin bir sonuca ulaştırabilir (Zorbaz, 2007:90). Bu konuyu Çoban (2014) yaptığı çalışmasında şu şekilde örneklendirmektedir.

Tablo 12. Örnek cümle ve okunabilirlik seviyeleri

No	Cümle	Ateşman Okunabilirlik Seviyesi	Çetinkaya-Uzun Okunabilirlik Seviyesi
1	Ben, annem ve babam pazartesi günü Kahramanmaraş'a uzun süredir görüşemediğimiz abimi ziyarete gideceğiz.	Kolay	Bağımsız Okuma (5, 6, 7. sınıf)
2	Galat-ı meşhur fasih-i mehcurdan evladır.	Çok Kolay	Bağımsız Okuma (5, 6, 7. sınıf)

Tablo 12'de görüldüğü üzere 1. cümle Ateşman okunabilirlik formülüne göre “Kolay”, Çetinkaya-Uzun okunabilirlik formülüne göre “Bağımsız Okuma” seviyesinde tespit edilmesine rağmen 2. cümle Ateşman okunabilirlik formülüne göre “Çok Kolay”, Çetinkaya-Uzun okunabilirlik formülüne göre “Bağımsız

Okuma” seviyesinde tespit edilmiştir. Nitekim Çoban (2014), yaptığı derleme çalışmasında bu örneği vererek; okunabilirlik formüllerinden çıkan mantıksal sonucun kelime ve cümle uzunluklarını temel alan formüllere göre cümleyi oluşturan kelimelerin çok olması ve kullanılan kelimelerin hece sayısının yüksek olması daha düşük kelimeli cümlelere ve hece sayısı daha düşük kelimelere göre okunabilirlik açısından daha zor olduğudur. Bu mantığa göre Tablo 12’deki örnekte 1. cümlenin 2. cümleye göre okunabilirlik açısından daha zor olması gerektiğini, fakat 1. cümlenin 2. cümleden uzun olması onu daha anlaşılabilir sağlamadığını belirterek “bir metnin okunabilirliğinin sadece sayısal verilerle ölçülemeyeceğini, metni oluşturan kelimelerin anlamlarının bilinip bilinmemesine göre okunabilirlik üzerinde etkili olduğu” düşüncesini belirginleştirmiştir.

Eğitsel içerikli web sayfaları ile ilgili Eraslan, 2008 yılında “Eğitsel içerikli web sitelerinin okunabilirlik açısından incelenmesi” adlı yüksek lisans tezinde web sayfalarındaki metinlerin okunabilirlik seviyesi ve fiziksel özellikleri açısından incelemeye almıştır. Metinleri okunabilirlik açısından Ateşman’ın Türkçe okunabilirlik formülü ile seviyesini belirlemeye çalışırken web sitelerindeki metin tasarım özelliklerini incelemek içinse hazırlanan 55 maddelik “Web sitesi metin tasarım unsurları değerlendirme ölçeği”, öğretim teknolojisi bölümünden 5, Türkçe eğitim alanından 3 uzman olmak üzere toplam 8 uzman değerlendirilmesinden geçerek hazırlanmıştır. Beş ayrı başlık altında toparlanan ölçek maddeleri: genel metin tasarımı, başlıklar, listeler, bağlantılar ve içerik metinleridir.

Araştırmanın kapsamı, Google arama motorunda “eğitsel içerikli web siteleri, eğitim siteleri, ders anlatımı, çevrimiçi ders, ders siteleri, uzaktan eğitim, öğretmen siteleri, çevrimiçi öğrenme, çocuk siteleri” ve benzeri anahtar kelimeleri ile yapılan arama sonucu ulaşılan eğitsel içerikli on web sitesidir.

Bu çalışmanın amacı Ateşman okunabilirlik formülü ve hazırlanan ölçek arasındaki anlamlı bir ilişkinin olup olmadığıdır. Cevap aranan sorular ise:

“Eğitsel içerikli web sitelerinin metin tasarım unsurları ölçeğinde her bir maddeden aldığı puan nedir?”, “Eğitsel içerikli web sitelerinin metin tasarım

unsurları ölçeğinde en çok dikkate alınan özellikler nelerdir?”, “Eğitsel içerikli web sitelerinin metin tasarım unsurları ölçeğinde en az dikkate alınan özellikler nelerdir?”, “Eğitsel içerikli web sitelerinde yer alan metinlerin, metin tasarım unsurları ölçeğine göre zorluk düzeyleri nedir?”, “Eğitsel içerikli web sitelerinde yer alan metinlerin okunabilirlik formülünden alınan sonuca göre okunabilirlik düzeyi nedir?”, “Web sitelerinin okunabilirlik formüllerine göre belirlenen okunabilirlik düzeyi ile metin tasarım unsurları ölçeği ile belirlenen düzeyleri arasında anlamlı bir ilişki var mıdır?” şeklindedir (Eraslan, 2008).

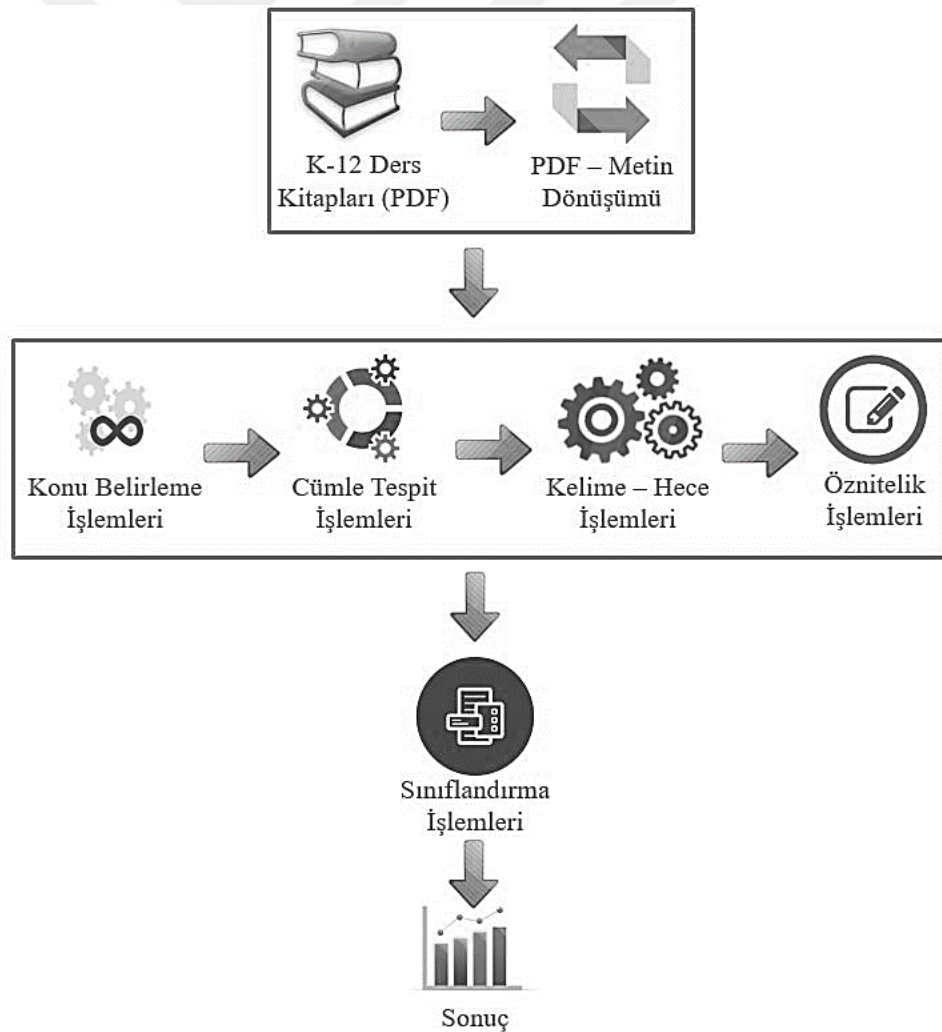
Sonuç olarak ölçek seviyeleri ile okunabilirlik seviyelerinin benzerlik gösterdiği, bazı farklılıkların ise ölçeğin 3’lü derecelemeye, okunabilirliğin 5’li derecelemeye göre değerlendirilmesinden kaynaklandığı belirtilmiştir.

Elde edilen veriler doğrultusunda Türkiye’de yayınlanan web sitelerinin okunabilirlik ve metin tasarımı açısından seviyeleri hakkında yorum yapılmış, değerlendirilen 10 sitenin dağılım frekansı göz önünde bulundurularak Türkiye’de yayınlanan web sitelerinden %20’si okunabilirlik açısından kolay, %50’si orta güçlükte, %30’u ise zor olarak nitelendirilmiştir (Eraslan, 2008).

İKİNCİ BÖLÜM

EĞİTSEL İÇERİKLİ VERİLERİN SINIFLANDIRILMASI

Bu çalışmadaki amacımız, eğitsel içerikli web sayfalarının okunabilirlik seviyelerine göre kademe bazlı (ilkokul, ortaokul ve lise) veya sınıf bazında (1. sınıf, 2. sınıf gibi) otomatik sınıflandırılmasıdır. Bu amaçla genel yöntemimiz K-12 ders kitaplarının öğrenme kümesi olarak kullanılması ve bu kitaplardan literatürdeki okunabilirlik çalışmalarını da göz önüne alarak özniteliklerin çıkarılmasına dayalıdır. Çıkarılan öznitelikler ile sınıflandırma modelleri öğrenilmektedir. Şekil 1’de bu genel yöntemimiz gösterilmiştir.



Şekil 1. Eğitsel içerikli verilerin analizindeki süreçler

Eğitsel içerikli verileri sınıflandırırken eğitim ve test olmak üzere iki farklı veri seti kullanılmıştır. Eğitim verileri olarak Milli Eğitim Bakanlığınca yayınlanan ilkokul, ortaokul ve lise kademelerindeki ders kitapları kullanılmıştır. Bu kitaplar Milli Eğitim Bakanlığı Talim ve Terbiye Kurulu Başkanlığınca incelenmektedir. Başkanlığın “Taslak ders kitaplarının incelenmesinde, değerlendirmeye esas olacak kriterler” konulu çalışması göre incelenen kitapların öğrenci seviyesine uygun olup olmadığı değerlendirilmektedir. Öğrenci seviyelerine uygunluk çalışması ilgili başkanlıkça yapıldığı için çalışmamızda, yayınlanmış bu kitapların kendi sınıf seviyelerine uygun olduğu kabul edilmiştir. Test verileri ise eğitsel içerikli internet verilerinden elde edilmiştir.

Mevcut okunabilirlik formüllerinin birçoğunda olduğu gibi, bu tez çalışmasında kullanılan Ateşman, Çetinkaya-Uzun ve ARI Okunabilirlik formüllerinin hesaplanabilmesi için harf, hece, kelime ve cümle sayılarının belirlenmesi gerekmektedir.

Okunabilirlik formüllerinin yanı sıra yeni öznitelikler üzerinde çalışmalar yapıp okunabilirlik seviyelerine göre sınıflandırma işlemine sağladığı katkılar incelenmiştir.

2.1. VERİ DÖNÜŞÜM VE HAZIRLAMA İŞLEMLERİ

Milli Eğitim Bakanlığının resmi sitesinden indirilen veriler pdf (portable document format) dosya biçiminde bulunmaktadır. Verilerin işlenebilir hale getirilmesi için metin dosya biçimine dönüştürülmesi gerekmektedir. Dönüşüm işleminde pdf dosyalarında bulunan resimler nedeniyle bazı yazılar eksik veya hatalı çevrilebilmektedir. Bu nedenle dönüşüm işleminden sonra metin dosyalarında hatalar kontrol edilerek düzeltilmiştir.

2.2. KONU BELİRLEME İŞLEMLERİ

Ders kitapları ve internetten elde edilen verilerden öğrenme ve test kümeleri oluşturulurken metinlerdeki her bir konu ve alt konu bir örnek (instance) olarak

kabul edilmiştir. Bu nedenle metinlerin bölümlenebilmesi için elle, konu ve alt konu ayrıçaları konulmuştur.

Metin dosyalarında tek konu, birden fazla konu veya konu ile ilişkili alt konular olabilmektedir. Bu nedenle metinlerdeki konular ve alt konular ayrıçalar ile işaretlenerek bir konunun nerede başlayıp nerede bittiği belirtilmiştir. Alt konuların belirtilmesi sayesinde konuların içerisinde bulunan alt başlıkların da farklı bir konu gibi anlaşılması sağlanmış, bununla birlikte bağlı bulunduğu konunun bir alt konusu olduğu belirtilmiştir.

2.3. CÜMLE TESPİT İŞLEMLERİ

Verilerde bulunan her bir örnek için cümle tespit çalışması yapılmıştır. Cümle tespiti için Aktaş (2006) ve Dinçer'in (2004) çalışmaları incelenerek, çalışmalarındaki cümle ayrıçaları, cümle kuralları ve cümle tespit algoritması kullanılmıştır. Ayrıca algoritmada Türk Dil Kurumu (TDK) tarafından yayınlanan kısaltmalar dizini listesi de kullanılmaktadır. Bu sayede kısaltmalarda bulunan nokta işareti cümle sonu olarak algılanmamaktadır. Aktaş (2006) geliştirdiği algoritmasında, "Türkçe cümle sonları, önceden belirlenen kural ve kısaltmalar listeleri kullanılarak %98.96 oranında başarıyla" tespit edildiğini belirtmiştir. Bu tez çalışmasında belirtilen cümle tespit algoritması kullanılarak her bir örnekteki cümleler tespit edilmiş ve böylece okunabilirlik formülleri için cümle sayıları tespit edilmiştir.

2.4. KELİME-HECE İŞLEMLERİ

Kelime işlemlerinde, metinlerdeki her bir kelimenin hecelerine ve köklerine ayırma işlemleri gerçekleştirilmektedir. Bu işlemler için açık kaynak kodlu Türkçe doğal dil işleme kütüphanesi olan Zemberek kullanılmıştır (Zemberek, t.y.). Bu sayede kelimelerin harf sayısı, hece sayısı ve kelime kökü bilgisi elde edilmiştir.

2.5. ÖZNETELİK İŞLEMLERİ

Verilerimizde sınıflandırma işlemi yapabilmek için metinlerden elde edilen, sınıflandırmada doğruluk oranına katkı sağlayacak öznitelikler bilgisine ihtiyaç

duyulmaktadır. Mevcut formüller kullanılarak Ateşman, Çetinkaya-Uzun ve ARI okunabilirlik formülleri sonuçları birer öznitelik olarak kullanılmıştır.

Mevcut formüllerin yanı sıra, bu tez kapsamında yeni özniteliklerin bulunması ve sınıflandırmaya etkisi de incelenmiştir. Bu bağlamda yaş seviyelerine göre kelime listeleri, detaylı hece sayıları ve okunabilirlik formüllerinde olduğu gibi, hesaplanan sayısal değerlerin birbirlerine oranı yeni öznitelik olarak kullanılmıştır. Bu öznitelikler detaylı olarak alt bölümlerde anlatılmıştır.

2.5.1. Kelime Listesi Öznitelikleri

Okunabilirlik formüllerinde anlamlarına göre sınıflandırılmış kelime listeleri kullanılabilir. Bu çalışmada, Keklik'in (2010, 2011) yaptığı araştırmaya göre belirlediği, Türkçede 0-6 yaş çocuklarına öğretilmesi gereken, en sık kullanılan 1200 kelime ve 7-11 yaş çocuklarına öğretilmesi gereken, en sık kullanılan 1800 kelime kullanılmıştır. Bu kelime listeleri ile metinlerdeki kelimelerin yüzde olarak kaçının 0-6 yaş kelimesi veya 7-11 yaş kelimesi olduğu bilgisi, iki ayrı öznitelik şeklinde ifade edilmiştir.

2.5.2. Hece Öznitelikleri

Hece sayıları okunabilirlik formüllerinin büyük bir kısmında kullanılan parametrelerden birisidir. Türkçe için yapılan çalışmalarda elde edilen sonuçlara göre ortalama kelime uzunluğu 2.60 hecedir (Ateşman, 1997; Bezirci, 2010). Farklı çalışmalar ve formüllerde hece sayının kullanılması nedeniyle hece sayılarına göre öznitelikler tanımlanarak sınıflandırmaya etkisi incelenmiştir.

Tez çalışmasında kullanılan hece öznitelikleri: bir heceli, iki heceli, üç heceli, dört heceli, beş heceli, altı heceli, yedi ve üzeri heceli kelimelerin yüzde olarak hesaplanmasıyla yedi farklı öznitelik şeklinde ifade edilmiştir.

2.5.3. Oran Öznitelikleri

Okunabilirlik formüllerinde kullanılan harf, hece, kelime ve cümle sayısının birbirlerine oranlarının ayrı ayrı incelenebilmesi için farklı öznitelikler olarak ifade edilerek sınıflandırmaya etkisi incelenmiştir.

Tez çalışmasında incelenen oran öznitelikleri: harf hece oranı, harf kelime oranı, harf cümle oranı, hece kelime oranı, hece cümle oranı, kelime cümle oranı olarak altı farklı öznitelik şeklinde ifade edilmiştir.

2.6. SINIFLANDIRMA İŞLEMLERİ

Verilerimizden tespit edilen özniteliklerden farklı kombinasyonlarda veri setleri üretilerek 22 farklı sınıflandırma algoritması ile analiz edilerek yeni modeller oluşturulmuştur. Bu modeller test verileri ile test edilerek sınıflandırma modellerinin başarı oranları tespit edilmiştir.

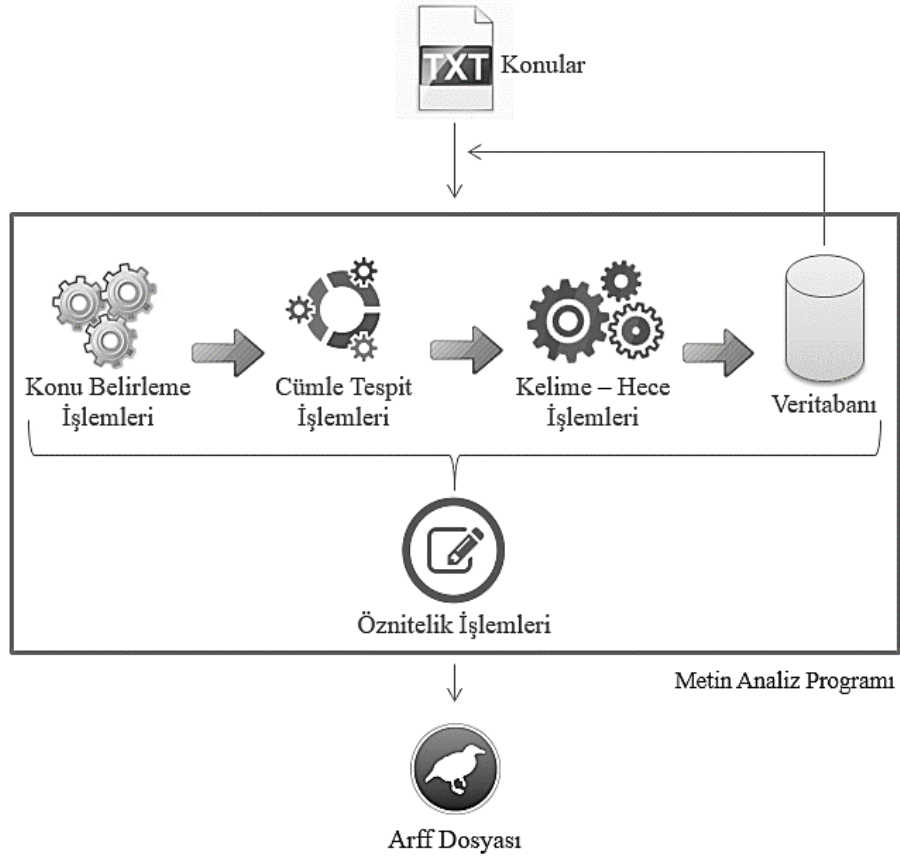
2.7. METİN ANALİZ PROGRAMI VE KULLANILMASI

Bu programı C# programlama dili kullanılarak tez kapsamında geliştirilmiş olup verilerin cümle, kelime ile hece ve öznitelik işlemlerinin yapılmasında kullanılmıştır. Programın çalışma mantığı Şekil 2’de gösterilmiştir. Program, ilişkisel veri tabanı ile kullanılmakta olup yapılan analizlerin sonuçları veri tabanına kaydedilmektedir.

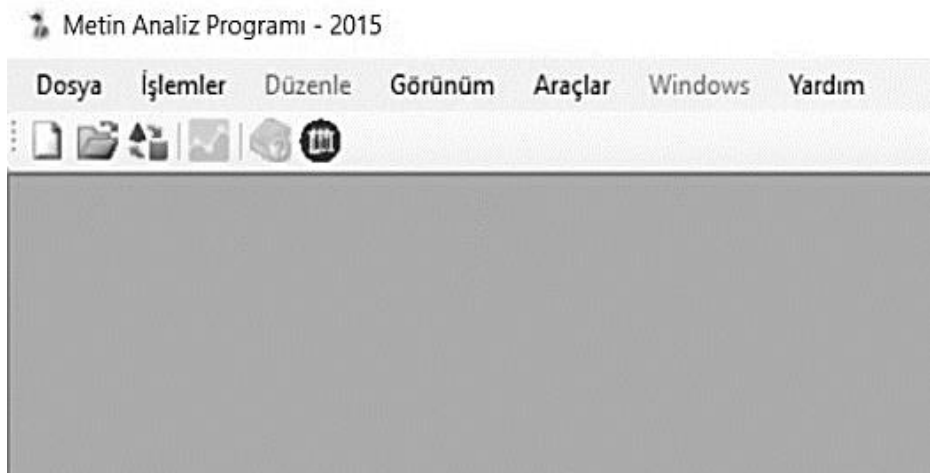
Temel olarak bu programın dört amacı vardır. Bunlar:

- Metinlerdeki konuların belirlenmesi,
- Metinlerdeki cümlelerin tespit edilmesi,
- Metinlerde tespit edilen cümlelerdeki kelimelerin hece ve köklerinin tespit edilmesi,
- Tez kapsamında yeni özniteliklerin hesaplanmasıdır.

Programın açılış ekranı Şekil 3’te gösterilmiştir.



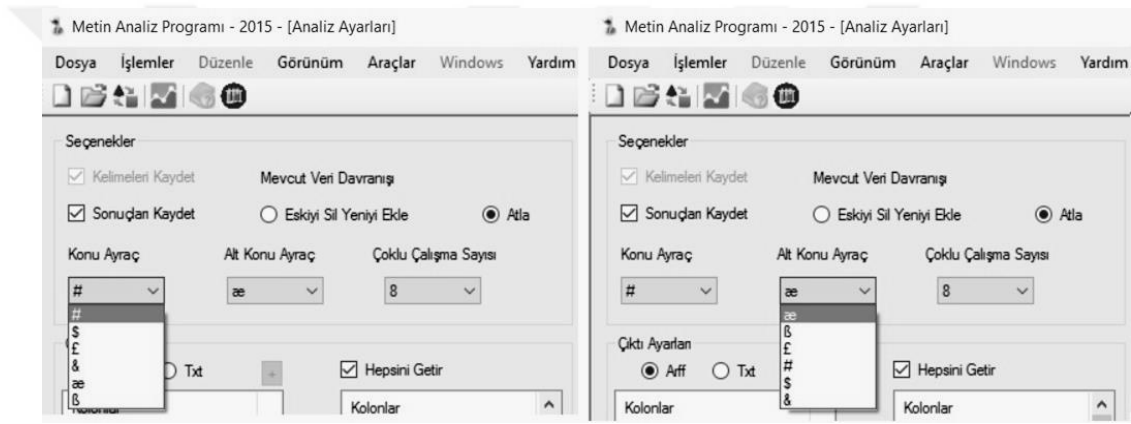
Şekil 2. Metin analiz programı iş akış diyagramı



Şekil 3. Metin analiz programı giriş ekranı

2.7.1. Veri Giriş-Çıkışı ve Konuların Belirlenmesi

Metin Analiz Programında veri girişi metin dosyası şeklindedir. Tek bir dosya olabileceği gibi birden çok dosya olarak da veri girişi sağlanabilmektedir. Metin dosyalarında tek konu, birden fazla konu veya konu ile ilişkili alt konular olabilmektedir. Bu nedenle metinlerdeki konular ve alt konular, programın anlayabilmesi için ayraç ile belirtilmelidir. Ayraç olarak kullanılabilen özel karakterler programda sabitlenmiştir. Şekil 4’ te gösterilen bu karakterler: #, \$, £, &, æ ve ß şeklindedir.



Şekil 4. Metin analiz programı konu ve alt konu ayraçları

Konularına ve alt konularına ayrılmış metinler programa verildiğinde, belirtilen ayraçlarla parçalara ayrılır.

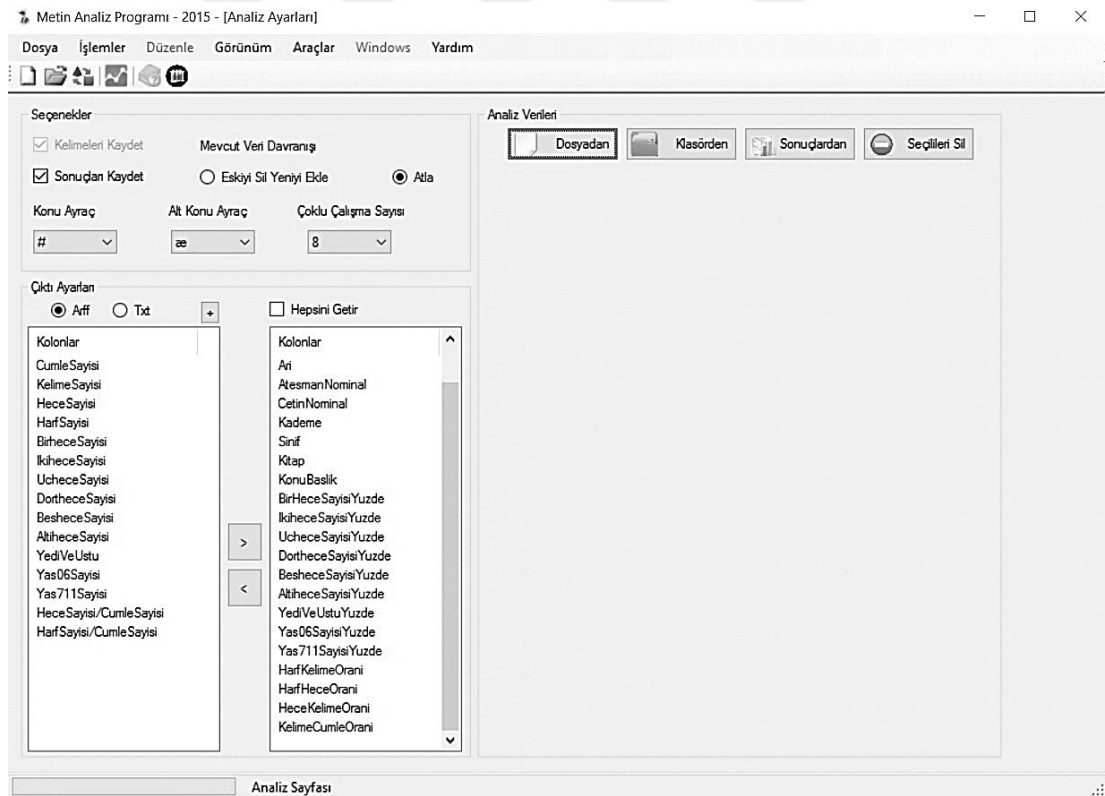
Tüm analiz işlemleri sonucunda verilerin sınıflandırma programının işleyebileceği, uygun bir formatta çıktı olarak sunulması gerekmektedir. Bu nedenle çıktı formatı arff (attribute-relation file format) dosya biçimi olarak belirlenmiştir. Arff dosya biçimi, sınıflandırma programının giriş (input) dosya biçimidir.

2.7.2. Metin Analiz Programının Kullanılması

Metin Analiz programı yeni analiz işlemi, veri tabanından sorgulama işlemleri, verilerin veri tabanına aktarılması (import) veya dışarı kaydedilmesi (export) işlemleri ile normalizasyon işlemi şeklinde fonksiyonel olarak dört ana başlıkta toplanmıştır.

2.7.2.1. Yeni Analiz Fonksiyonu

Program açıldıktan sonra kısa yol araç çubuğundan veya *Dosya* menüsünde bulunan *Yeni Analiz* butonu ile yeni analiz ekranı çalıştırılır. Şekil 5'te yeni analiz ekranı çalıştırıldığında ekran görüntüsü gösterilmektedir. Sağ tarafta bulunan analiz verileri seçeneklerinde veri girişi için kullanılabilen butonlar bulunmaktadır. Veri girişi dosyadan, klasörden veya veri tabanında kayıtlı analiz sonuçları olmak üzere üç farklı şekilde veri seçimi gerçekleştirilebilmektedir. Veri seçimi dosyadan veya klasörden seçilirse, verilerin sınıf seviyelerini zorunlu olarak belirtmemiz gerekmektedir. Kitap isimleri ise varsayılan dosya adı olacak kitap adı kısmına gelmekte ve tercihe göre değiştirilebilmektedir. Veri seçiminde mevcut sonuçlardan seçim yaptysak, veri tabanındaki hâlihazırdaki bilgiler kullanıldığından sınıf ve kitap adı gibi bilgiler veri tabanından alınmaktadır.



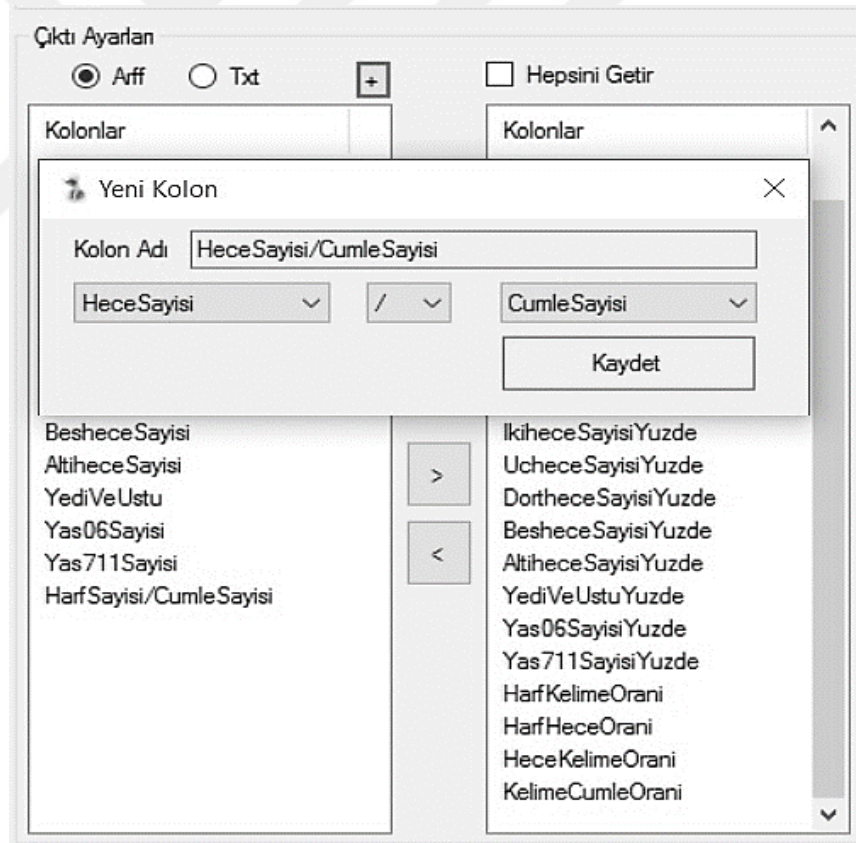
Şekil 5. Metin analiz programı yeni analiz ekranı

Yeni analiz ekranında seçenekler bölümünde işlem süreciyle ilgili detaylar bulunmaktadır. Sonuçların kaydedilmesi, mevcut veri varsa eskisinin yerine mi

ekleneceği veya işlem yapılmadan geçileceği gibi seçeneklerin yanı sıra, konu ve alt konu ayraç belirleme ile paralel çalışabilme seçenekleri bulunmaktadır.

Program analiz işleminde daha kısa sürede işlemleri tamamlayabilmek için paralel çalışabilme mantığında geliştirilmiştir. Bu nedenle program bilgisayardaki CPU bilgisine bakarak bilgisayarın özelliğine göre çoklu çalışabilme seçeneği sunmaktadır. Burada seçilen değer sayısında iş parçacığı oluşturularak analiz edilen metinlerin bu iş parçacıklarına sıra ile dağıtılması sağlanır ve bu iş parçacıkları paralel çalıştığından işlem süresinde gözlenebilir farklar oluşturabilmektedir.

Analizin sonucunun hangi özniteliklerden oluşacağını çıktı ayarları kısmında belirlenebilmektedir. Şekil 6'da çıktı ayarları kısmı gösterilmiştir.

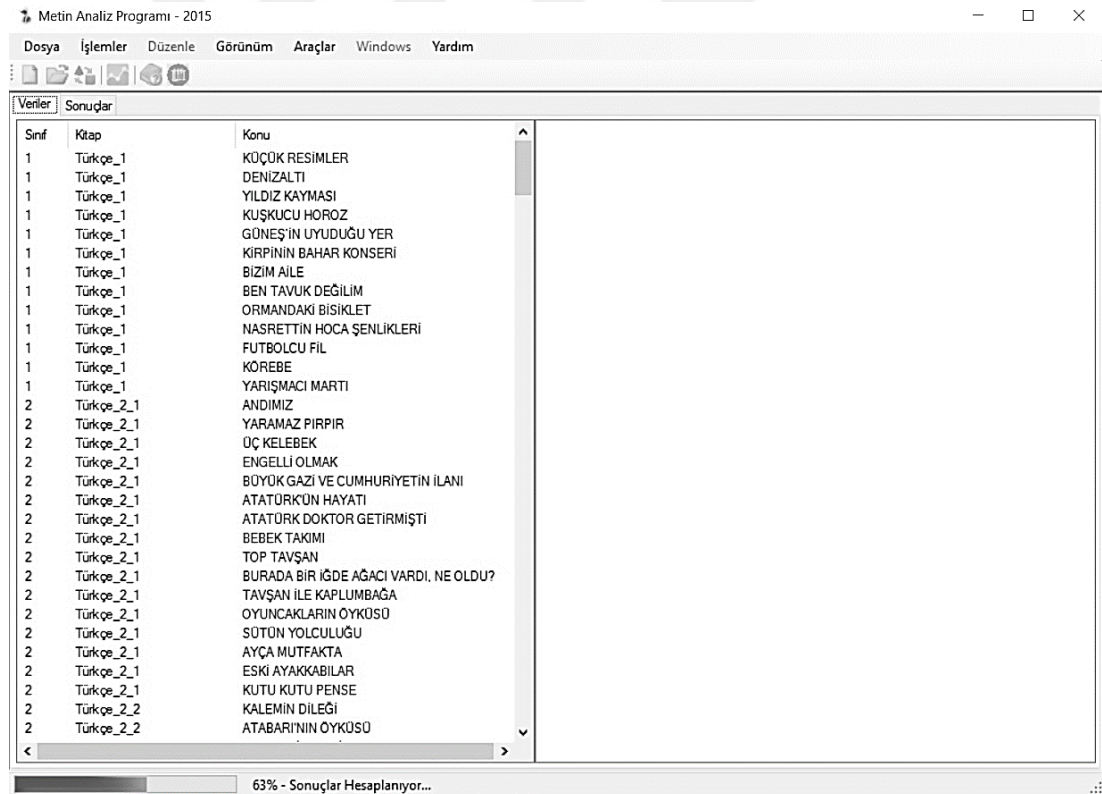


Şekil 6. Metin analiz programı çıktı ayarları

İlk olarak çıktının tipi belirtilmelidir. Eğer analiz çıktısı Weka programı ile analiz edilmek isteniyorsa çıktı ayarları kısmından, Weka programı dosya biçimi

olan *Arff* seçilmelidir. Weka programı java programlama dili ile geliştirilmiş, arff dosya biçimindeki verileri inceleyebilen, içerisinde birçok makine öğrenme algoritmalarını barındıran bir kütüphanedir (Hall ve diğ., 2009). Eğer metin dosyası tipinde virgülle ayrılmış veri isteniyorsa *Txt* seçeneği ile bu şekilde sonuçlar alınabilmektedir. Öznitelik seçim ekranında çıktıda bulunması istenilen alanlar ayarlanabilmektedir. Mevcut kolonlardan istenilen kolonun eklenip çıkarılabileceği gibi mevcut sayısal kolonlar kullanılarak yeni kolon da türetilmektedir. Yeni kolon türetme işlemi Şekil 6’da gösterilmiştir.

İstenilen ayarlar tamamlandıktan sonra kısa yol araç çubuğundan veya *İşlemler* menüsünden *Analiz Et* butonu ile analize başlanır. Analiz işlemine başlandığında Şekil 7’deki ekran görünür.



Sınıf	Kitap	Konu
1	Türkçe_1	KUÇUK RESİMLER
1	Türkçe_1	DENİZALTI
1	Türkçe_1	YILDIZ KAYMASI
1	Türkçe_1	KUŞKUCU HOROZ
1	Türkçe_1	GÖNEŞ'İN UYUDUĞU YER
1	Türkçe_1	KIRPININ BAHAR KONSERİ
1	Türkçe_1	BİZİM AİLE
1	Türkçe_1	BEN TAVUK DEĞİLİM
1	Türkçe_1	ORMANDAKI BİSİKLET
1	Türkçe_1	NASRETTİN HOCA ŞENLİKLERİ
1	Türkçe_1	FUTBOLCU FIL
1	Türkçe_1	KÖREBE
1	Türkçe_1	YARIŞMACI MARTI
2	Türkçe_2_1	ANDİMİZ
2	Türkçe_2_1	YARAMAZ PIRPIR
2	Türkçe_2_1	UÇ KELEBEK
2	Türkçe_2_1	ENGELLİ OLMAK
2	Türkçe_2_1	BÜYÜK GAZİ VE CUMHURİYETİN İLANI
2	Türkçe_2_1	ATATÜRKÜN HAYATI
2	Türkçe_2_1	ATATÜRK DOKTOR GETİRİŞTİ
2	Türkçe_2_1	BEBEK TAKIMI
2	Türkçe_2_1	TOP TAVŞAN
2	Türkçe_2_1	BURADA BİR İĞDE AĞACI VARDI, NE OLDU?
2	Türkçe_2_1	TAVŞAN İLE KAPLUMBAĞA
2	Türkçe_2_1	OYUNCAKLARIN OYKÜSÜ
2	Türkçe_2_1	SÜTÜN YOLCULUĞU
2	Türkçe_2_1	AYÇA MUTFAKTA
2	Türkçe_2_1	ESKİ AYAKKABILAR
2	Türkçe_2_1	KUTU KUTU PENSE
2	Türkçe_2_2	KALEMİN DİLEĞİ
2	Türkçe_2_2	ATABARININ OYKÜSÜ

Şekil 7. Metin analiz programı sonuçların hesaplanması

Analiz işlemi başladığında ilk olarak veri kaynağına göre metinlerden veya veri tabanındaki sonuçlardan veriler alınır. Metin dosyası için belirtilen konu ve alt

konu ayraçları kullanılarak dosyalar kitap, konu başlığı ve metni olarak listelenir. Bu listeler belirtilen çoklu çalışma sayısına göre eşit olacak şekilde dağıtılır. Çoklu çalışma sayısınca oluşturulan iş parçacıklarına oluşturulan listeler verilerek paralel bir şekilde bu listelerin analizlerinin yapılması sağlanır.

Her bir iş parçacığı projede oluşturulan bir konsol uygulamasını dışardan çalıştırarak programda kullanılan cümle tespit algoritması yardımıyla metinlerin cümlelerinin tespit edilmesi sağlanır. Cümlelerdeki kelimelerin hecelenmesi için de zemberek kütüphanesi kullanılır. Bu konsol uygulamasının yazılma amacı, Zemberek kütüphanesinin kullandığı dosyaların tek bir iş parçacığında kullanılabilir olmasıdır. Oluşturulan iş parçacıkları sayısında zemberek kütüphanesinin kaynak dosyaları çoğaltılarak her bir iş parçacığının kendi kaynak dosyasının kullanılması sağlanmış ve böylelikle metin analiz programı paralel çalışabilir bir tasarıma getirilmiştir.

Program analiz işlemlerini tamamladıktan sonra analizin sonuçlarının belirtilen kolonlara göre çıktılarını hazırlama işlemine başlamaktadır. Bu işlem bittikten sonra çıktı oluşturulur ve analiz ekranında sınıf, kitap ve konu olarak analiz edilen veriler sıralanmaktadır. Bu listedeki her bir satır bizim bir konumuzu ifade etmektedir. Bu satırların üzerine tıkladığında, yan tarafta bulunan listede seçilen konuya ait cümleler sıralı olarak gelmektedir.

Yapılan analiz neticesinde hazırlanan sonuçlar, Şekil 8’de gösterildiği gibi sonuçlar sekmesinde, analiz ekranında ayarlanan kolonlara göre gösterilmektedir.

2.7.2.2. Veri Sorgulama Fonksiyonu

Veri sorgulama ekranı kısa yol araç çubuğundan veya *Dosya* menüsünde bulunan *Veritabanı* butonu ile aktif hale getirilmektedir. Bu ekranın amacı, veri tabanında bulunan verilerin kontrolü, yapılan sorgulama sonucunda gelen verilerin ekranda incelenmesi ve csv (comma-separated values) dosya biçiminde kaydedilebilir olmasıdır. Şekil 9’da kaydedilen sonuçlar sorgulanarak veri tabanında bulunan kayıtlı veriler getirilmiştir.

Metin Analiz Programı - 2015

Dosya İşlemler Düzenle Görünüm Araçlar Windows Yardım

Veriler Sonuçlar

	Kademe	Sınıf	Kitap	KonuBaşlık	BirHeceSayisiYuzde	IkheceSayisiYuzde	UcheceSayisiYuzde	DortheceSayisiYuzde
▶	ilkogretim	1	Türkçe_1	KUÇUK RESİMLER	0.11	0.15	0.10	0.06
	ilkogretim	1	Türkçe_1	DENİZALTI	0.07	0.15	0.11	0.04
	ilkogretim	1	Türkçe_1	YILDIZ KAYMASI	0.10	0.11	0.10	0.06
	ilkogretim	1	Türkçe_1	KUŞKUCU HOROZ	0.07	0.14	0.12	0.05
	ilkogretim	1	Türkçe_1	GÜNEŞİN UYUDUĞU YER	0.09	0.16	0.11	0.05
	ilkogretim	1	Türkçe_1	KIRPININ BAHAR KONSERİ	0.06	0.13	0.10	0.07
	ilkogretim	1	Türkçe_1	BİZİM AİLE	0.08	0.13	0.12	0.05
	ilkogretim	1	Türkçe_1	BEN TAVUK DEĞİLİM	0.09	0.17	0.10	0.05
	ilkogretim	1	Türkçe_1	ORMANDAKI BISİKLET	0.06	0.15	0.09	0.07
	ilkogretim	1	Türkçe_1	NASRETTİN HOCA ŞENLİKLERİ	0.06	0.07	0.11	0.07
	ilkogretim	1	Türkçe_1	FUTBOLCU FIL	0.08	0.15	0.11	0.05
	ilkogretim	1	Türkçe_1	KÖREBE	0.07	0.11	0.08	0.08
	ilkogretim	1	Türkçe_1	YARIŞMACI MARTI	0.06	0.12	0.11	0.05
	ilkogretim	2	Türkçe_2_1	ANDİMİZ	0.05	0.09	0.16	0.04
	ilkogretim	2	Türkçe_2_1	YARAMAZ PIRPIR	0.06	0.15	0.14	0.04
	ilkogretim	2	Türkçe_2_1	ÜÇ KELEBEK	0.07	0.10	0.10	0.06
	ilkogretim	2	Türkçe_2_1	ENGELLİ OLMAK	0.04	0.06	0.09	0.06
	ilkogretim	2	Türkçe_2_1	BOYUK GAZİ VE CUMHURİYETİN İLANI	0.03	0.09	0.12	0.05
	ilkogretim	2	Türkçe_2_1	ATATÜRKÜN HAYATI	0.05	0.10	0.13	0.06
	ilkogretim	2	Türkçe_2_1	ATATÜRK DOKTOR GETİRMİŞTİ	0.07	0.12	0.10	0.07
	ilkogretim	2	Türkçe_2_1	BEBEK TAKIMI	0.08	0.13	0.10	0.06
	ilkogretim	2	Türkçe_2_1	TOP TAVŞAN	0.09	0.13	0.11	0.05
	ilkogretim	2	Türkçe_2_1	BURADA BİR İĞDE AĞACI VARDI, NE OLDU?	0.07	0.12	0.10	0.06
	ilkogretim	2	Türkçe_2_1	TAVŞAN İLE KAPLUMBAĞA	0.05	0.17	0.11	0.05

100% - Databaseden sonuçlar tamamlandı. Toplam Süre: 00:00:18

Şekil 8. Metin analiz programı sonuçlar sekmesi

Metin Analiz Programı - 2015

Dosya İşlemler Düzenle Görünüm Araçlar Windows Yardım

Tablolar SQL Sorgusu

Kelimeler
Sonuçlar
Yas_0_6_Kelimeleri
Yas_7_11_Kelimeleri

select * from Sonuclar

KOŞ

Sonuçlar

Sorgu başanlı.

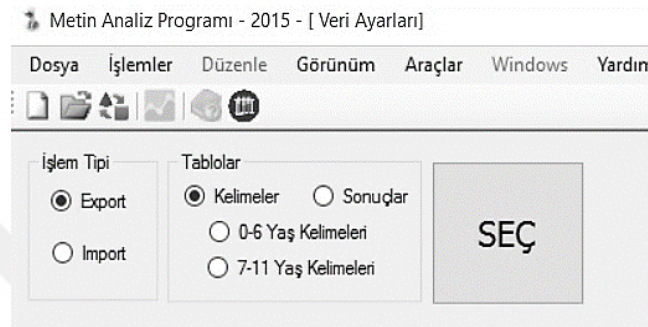
	Id	Kademe	Sınıf	Kitap	HeceSayisi	KonuBaşlık	CumleSayisi	KelimeSayisi	HarfSayisi	BirHe-
▶	956	ilkogretim	1	Türkçe_1	308	KUÇUK RESİMLER	19	133	741	35
	957	ilkogretim	1	Türkçe_1	348	DENİZALTI	19	138	803	25
	958	ilkogretim	1	Türkçe_1	433	YILDIZ KAYMASI	28	172	1010	42
	959	ilkogretim	1	Türkçe_1	518	KUŞKUCU HOROZ	27	205	1212	37
	960	ilkogretim	1	Türkçe_1	503	GÜNEŞİN UYUDUĞU YER	31	211	1170	45
	961	ilkogretim	1	Türkçe_1	653	KIRPININ BAHAR KONSERİ	35	247	1545	38
	962	ilkogretim	1	Türkçe_1	325	BİZİM AİLE	25	130	775	26
	963	ilkogretim	1	Türkçe_1	721	BEN TAVUK DEĞİLİM	50	305	1678	65
	964	ilkogretim	1	Türkçe_1	840	ORMANDAKI BISİKLET	65	323	2010	47
	965	ilkogretim	2	Türkçe_2_1	103	ANDİMİZ	7	37	251	5
	966	ilkogretim	1	Türkçe_1	458	NASRETTİN HOCA ŞENLİKLERİ	13	159	1105	27
	967	ilkogretim	1	Türkçe_1	632	FUTBOLCU FIL	49	253	1499	49
	968	ilkogretim	1	Türkçe_1	669	KÖREBE	31	244	1560	45
	969	ilkogretim	2	Türkçe_2_1	803	YARAMAZ PIRPIR	50	323	1901	49
	970	ilkogretim	2	Türkçe_2_1	641	ÜÇ KELEBEK	32	237	1512	46

Satır Sayısı: 603 | Süre: 00:00:00.1060444

Şekil 9. Metin analiz programı konu başlık veri tabanı sorgulama ekranı

2.7.2.3. Veri Al / Ver Fonksiyonu

Veri al / ver ekranı (Şekil 10) kısa yol araç çubuğu veya *Dosya* menüsünde bulunan *Veri Al / Ver* butonu ile aktif hale getirilmektedir. Bu fonksiyon veri tabanında bulunan verilerin yedeklenebilmesi ve gerekli olduğu durumlarda yedeklerin tekrar yüklenerek kullanılabilir hale getirilmesi amacıyla geliştirilmiştir.



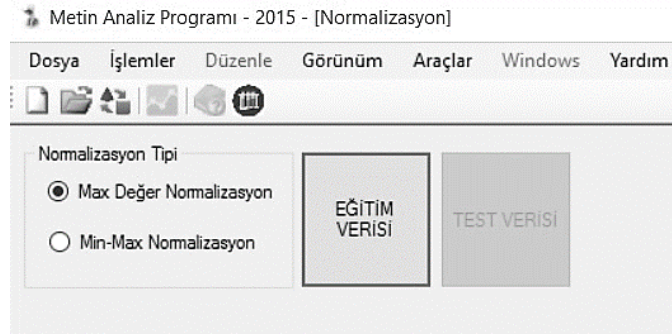
Şekil 10. Metin analiz programı veri al / ver ekranı

Verilerin dışarı aktarılması (export) işleminde seçilen tablodaki veriler csv dosya biçiminde istenilen yere kaydedilmektedir. Verilerin içeri aktarılması (import) işleminde ise seçilen tabloya, belirtilen csv biçimindeki dosyadan veriler alınarak aktarılmaktadır.

2.7.2.4. Normalizasyon Fonksiyonu

Normalizasyon ekranı kısa yol araç çubuğundan veya *İşlemler* menüsünde bulunan *Normalizasyon* butonu ile aktif hale getirilir. Açılan veri normalizasyon ekranı Şekil 11'de gösterilmiştir. Bu fonksiyonun amacı, arff dosya biçiminde hazırlanmış verilerin iki farklı normalizasyon işlemine göre normalize edilmesidir. Seçilen normalizasyon tipine göre eğitim verisi normalize edildiğinde aynı işlem ile test verisi de eğitim verisinde tespit edilen en küçük ve en büyük değerlere göre normalize edilebilmektedir.

Maksimum değer normalizasyon işleminde tüm değerler, aynı kümedeki en büyük elemana bölünerek listedeki elemanların en büyüğünün 1 olacak şekilde normalize edilmesini sağlar.



Şekil 11. Metin analiz programı normalizasyon ekranı

Minimum-maksimum normalizasyon işleminde tüm değerler kümesinde en büyük ve en küçük veriler tespit edilerek, her bir elemanın en küçük değerden çıkartılıp en büyük değere bölünmesi ile normalize edilmesi sağlanır. Bu sayede en küçük değer 0, en büyük değer 1 ve diğer değerler 0-1 aralığında olacak şekilde dağılım sağlanır.

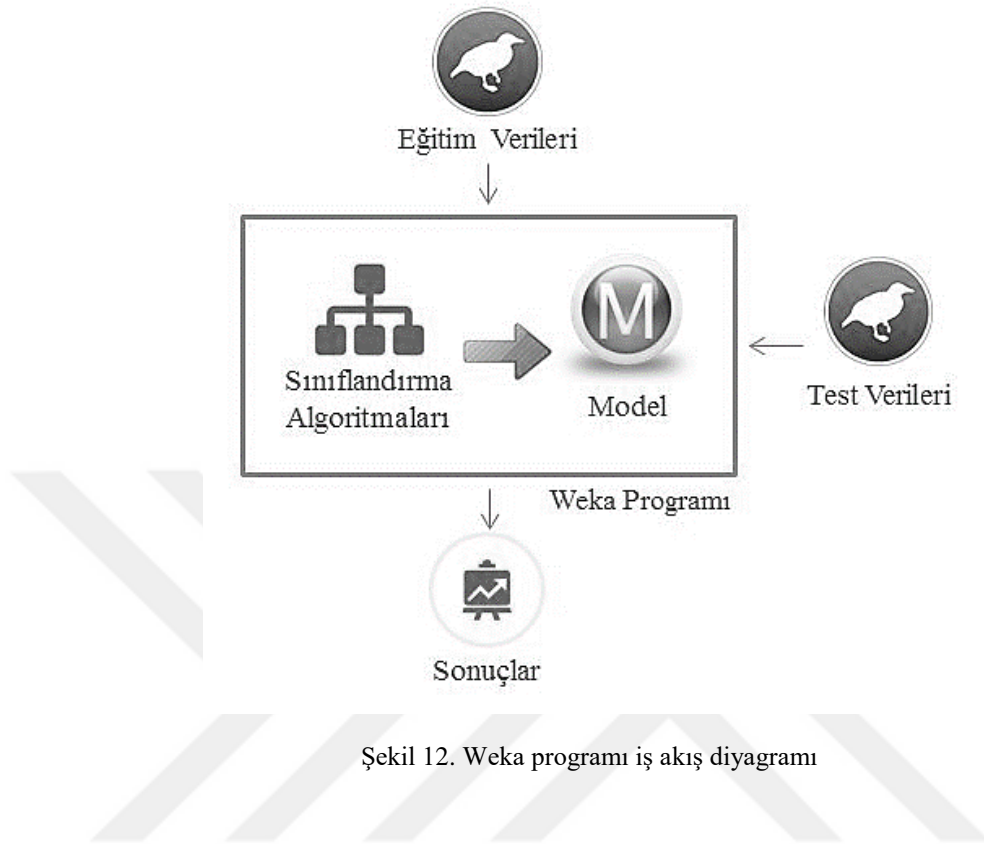
2.8. WEKA PROGRAMI

Bu tezde sınıflandırma işlemleri için java tabanlı açık kaynak kodlu bir makine öğrenmesi kütüphanesi olan Weka kullanılmıştır. Programın veri biçimi, arff formatında hazırlanmış verilerdir.

Weka programına hazırlanan veri girişi olarak eğitim ve test setleri verilir. Weka, içerdiği çok sayıda algoritmalarından seçilenler ile eğitim verisi üzerinde öğrenme işlemi yaparak her bir algoritma için modeller oluşturur. Verilen test verileri ile oluşturulan bu model test edilerek, modelin başarı oranı belirlenir. Bu iş akış diyagramı Şekil 12’de verilmiştir.

Weka programında kullandığımız başarı ölçüğü doğruluk oranıdır (accuracy). Bu oran, test verilerinin gerçek sınıflarına göre doğru tahmin edilen sayıların yüzde olarak sayısıdır.

Weka ile yapılan sınıflandırma işlemlerinde 22 farklı algoritma kullanılmıştır. Bu algoritmalar Tablo 13’te gösterilmiştir.



Tablo 13. Weka programı ile kullanılan 22 farklı algoritma listesi

AdaBoostM1(Default)	HoeffdingTree
AdaBoostM1 (HoeffdingTree)	J48
AdaBoostM1 (NaiveBayes)	LibSVM
AdaBoostM1 (SMO)	LogitBoost
AttributeSelecetedClassifier	MajorityVoting (NB_HT_SVM_SMO)
Bagging (Default)	MultilayerPerceptron
Bagging (HoeffdingTree)	NaiveBayes
Bagging (NaiveBayes)	RandomForest
Bagging (SMO)	SimpleLogistic
ClassificationViaRegression	SMO
FilteredClassifier	Stacking (NB_HT_SVM_SMO)

Weka ile 22 farklı algortmada işlemleri tek seferde yapabilmeye yarayan bilgi akışı (knowledge flow) işlevi kullanılmıştır. Bu işlemde sürükle bırak şeklinde çalışan bir ekran yardımı ile veri girişleri ve algoritmalar ayarlanabilmektedir. Bu sayede bir veri için tüm algoritmalar ile testler yapılarak sonuçları kaydedilmektedir.

ÜÇÜNCÜ BÖLÜM

DENEYSEL SONUÇLAR

3.1 VERİLER

Eğitim verileri, Milli Eğitim Bakanlığının resmi sayfasında yayımlanmış olup, 2014-2015 eğitim öğretim yılında kullanılan 1, 2, 3, 4, 5, 6, 7 ve 8. sınıf Türkçe ders kitabı ile 9, 10, 11 ve 12. sınıf Türk Dili ve Edebiyatı ders kitapları verileridir.

Test verileri ise Google arama motoru ile bir Türkçe eğitim arama motoru olan eğitim.com sitesi tarafından "... sınıf Türkçe konu anlatımı" ve "... sınıf okuma metni" aramaları (4. sınıf Türkçe konu anlatımı, 8. sınıf okuma metni gibi) tüm sınıflar için yapılarak çıkan sonuçlardan 14 farklı web sitesinden elde edilen eğitsel içerikli metinlerdir. Bu web siteleri: www.egitimhane.com, www.dersteknik.com, <http://egitimevreni.com>, www.derscalisiyorum.com.tr, www.dil-bilgisi.net, www.gercekogretmen.com, www.edebiyatogretmeni.org, www.etkinlikpaylas.com, www.muhteva.com, www.sorubak.com, <http://ilkokulogretmeni.com>, www.sinifogretmenim.com, www.okumametinleri.com, www.turkceciler.com siteleridir.

Eğitim verileri ilkökul kademesinde 111, ortaokul kademesinde 111 ve lise kademesinde 111 olmak üzere toplamada 333 farklı örnekten oluşmaktadır. Test verileri ise ilkökul kademesinde 90, ortaokul kademesinde 90 ve lise kademesinde 90 olmak üzere toplamada 270 farklı örnekten oluşmaktadır.

3.1.1. Veri Ön Hazırlık

Ders kitapları pdf biçiminde olduğu için txt biçimine çevirme işleminden geçirilmiştir. Bu işlemde metinlerde meydana gelen yazım hataları kontrol edilerek düzeltilmiştir. Test verileri arama motorları ile yapılan aramalar sonucunda web sayfaları içerisindeki ilgili metinler elle çıkarılarak metin dosyası formatında kaydedilmiştir.

Eđitim ve test verileri iin konu ve alt konu belirleme alıřması yapılarak konular iin “#” simgesi, alt konular iin de “æ” simgesi ayra olarak belirlenmiř ve Metin Analiz Programında analiz yaparken konu ve alt konu ayraları belirtildiđi řekilde seilmiřtir.

3.2. DENEYSEL SONULAR

Eđitim ve test verileri ile belirlenen zneliklerin farklı kombinasyonlarında veri setleri oluřturularak Weka programındaki 22 farklı algoritma ile deneyler yapılmıř ve ıkan tm sonular Ek 1’de bařarı oranına gre azalan sıralamada tablo halinde verilmiřtir.

Bu deney sonularına gre tablonun ilk sıralarına bakıldıđında Kademe bazlı sınıflandırma sonularının sınıf bazlı sınıflandırma sonularına gre daha iyi olduđu gzlemlenmiřtir. Deney sonularına gre kademe bazlı sınıflandırma sonularında en iyi sonu veren ilk beř algoritma seilmiřtir. Belirlenen en iyi algoritmalar: Bagging (HoeffdingTree), AdaBoostM1 (NaiveBayes), NaiveBayes, Bagging (NaiveBayes), MajorityVoting (NB_HT_SVM_SMO) řeklindedir.

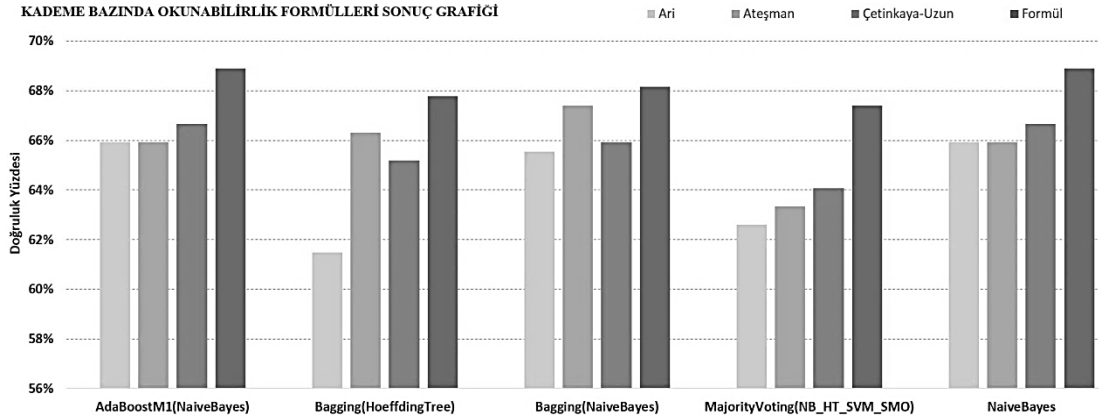
Yapılan deneyler kademe bazlı ve sınıf bazlı olmak zere iki kategoride toplanmıřtır. Her bir kategori iin eđitim ve test verilerimiz kullanılarak okunabilirlik formlleri zerine deneyler ve yeni oluřturulan znelikler zerine deneyler, belirlenen en iyi beř algoritmanın rettiđi sonulara gre yorumlanmıřtır.

3.3. KADEME BAZLI SONULAR

3.3.1. Ateřman, etinkaya-Uzun ve ARI Okunabilirlik Formlleri Sonuları

Bu deneyde Ateřman, etinkaya-Uzun ve ARI okunabilirlik formlleri ve  formln birlikte kullanıldıđı (forml zneliđi) veri setleri ile deneyler yapılmıřtır.

Ateřman, etinkaya-Uzun ve ARI okunabilirlik formllerinin ayrı ayrı test sonuları ile hepsinin birleřik test sonuları (forml zneliđi) kademe bazlı olarak řekil 13’te verilmiřtir.



Şekil 13. Kademe bazında okunabilirlik formülleri sonuç grafiği

Kademe bazlı sınıflandırma sonucuna göre en iyi başarı oranı, AdaBoostM1 (NaiveBayes) ve NaiveBayes algoritmaları Formül özniteliği için %68.9 doğruluk olarak elde edilmiştir.

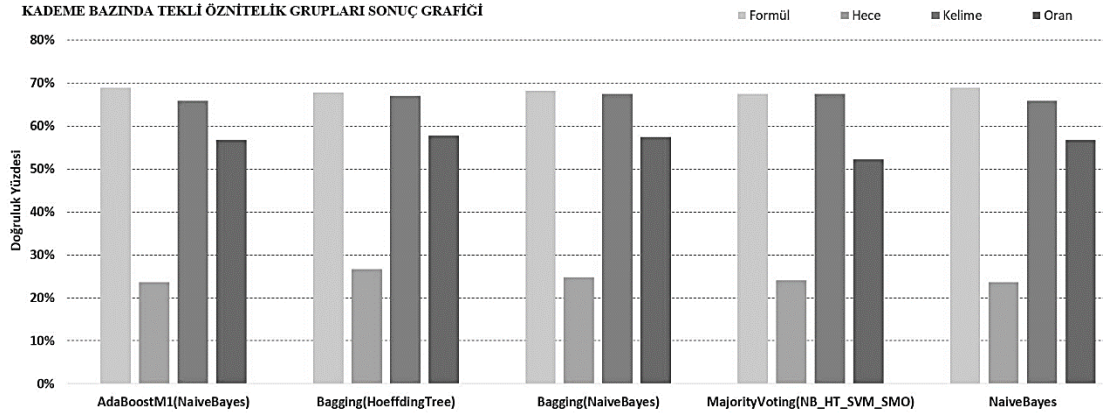
En iyi beş algoritmanın ortalama başarı oranlarına göre okunabilirlik formülleri arasındaki büyükten küçüğe sıralama, Ateşman, Çetinkaya-Uzun ve ARI olarak gözlemlenmiştir.

En iyi beş algoritmaya göre kademe bazlı yapılan Formül özniteliği; ARI, Ateşman ve Çetinkaya-Uzun okunabilirlik formüllerine göre daha başarılı sonuçlar ürettiği gözlemlenmiştir. Dolayısıyla okunabilirlik formüllerini tek tek değerlendirmek yerine Formül özniteliği grubu olarak değerlendirmeye alınmıştır.

3.3.2. Tekli Öznitelik Grubu Sonuçları

Bu deneyde Formül, Hece, Kelime ve Oran öznitelikleri veri setleri ile deney yapılmıştır. Kademe bazlı yapılan deney sonucu Şekil 14’de verilmiştir.

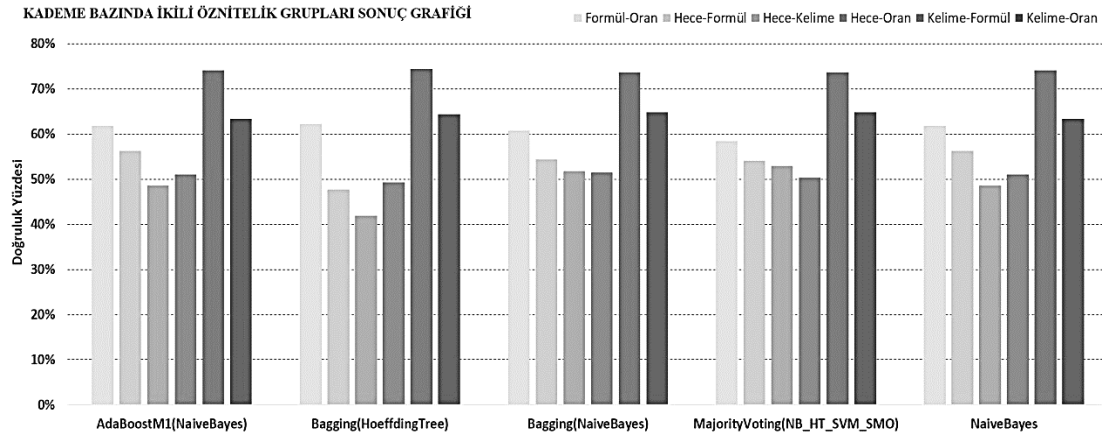
Kademe bazlı sınıflandırma sonucuna göre en iyi başarı oranı, AdaBoostM1 (NaiveBayes) ve NaiveBayes algoritmalarında Formül özniteliği için %68.9 olarak elde edilmiştir. Bu tez çalışmasında önerilen Kelime özniteliği ise Formül özniteliğine oldukça yakın (%67.4) doğruluk oranı elde etmiştir. Oran öznitelik grubu daha düşük başarı oranı (%57.8) yakalarken, Hece öznitelik grubu ise en düşük seviyede (%26.7) kalmıştır.



Şekil 14. Kademe bazında tekli öznitelik grupları sonuç grafiği

3.3.3. İkili Öznitelik Grubu Sonuçları

Bu deneyde Formül-Oran, Hece-Formül, Hece-Kelime, Hece-Oran, Kelime-Formül ve Kelime-Oran öznitelik grupları veri setleri ile deney yapılmıştır. Deney sonucunda çıkan sonuçlar kademe bazlı olarak Şekil 15'te verilmiştir.

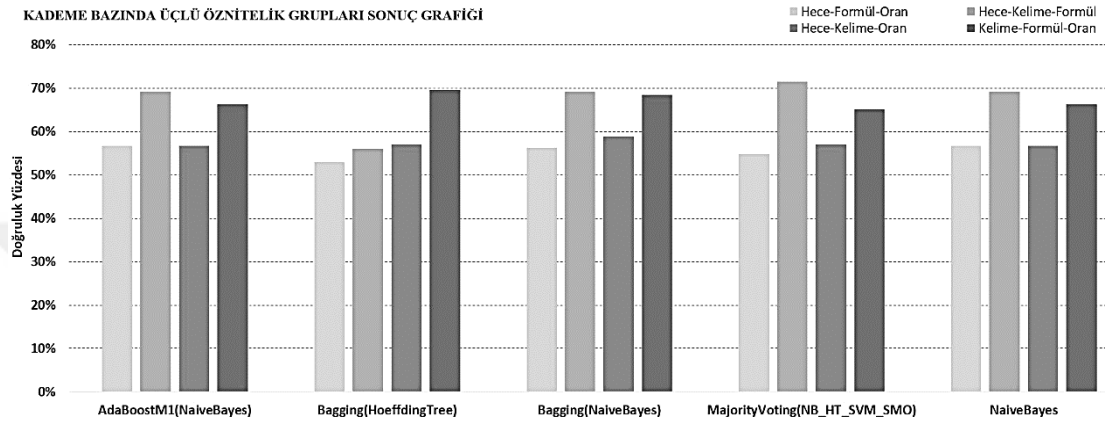


Şekil 15. Kademe bazında ikili öznitelik grupları sonuç grafiği

Kademe bazlı sınıflandırma sonucuna göre en iyi başarı oranı belirlenen en iyi beş algoritmada Kelime-Formül özniteliği için %74.4 olarak elde edilmiştir. Bu sonuca göre yeni özniteliklerden kelime öznitelikleri kullanılarak okunabilirlik formülleri ile elde edilen sonuçlardan daha iyi sınıflandırma sonucu elde edildiği görülmüştür.

3.3.4. Üçlü Öznitelik Grubu Sonuçları

Bu deneyde Hece-Formül-Oran, Hece-Kelime-Formül, Hece-Kelime-Oran, Kelime-Formül-Oran öznitelik grupları veri setleri ile deney yapılmıştır. Deney sonucunda çıkan sonuçlar kademe bazlı olarak Şekil 16’da verilmiştir.



Şekil 16. Kademe bazında üçlü öznitelik grupları sonuç grafiği

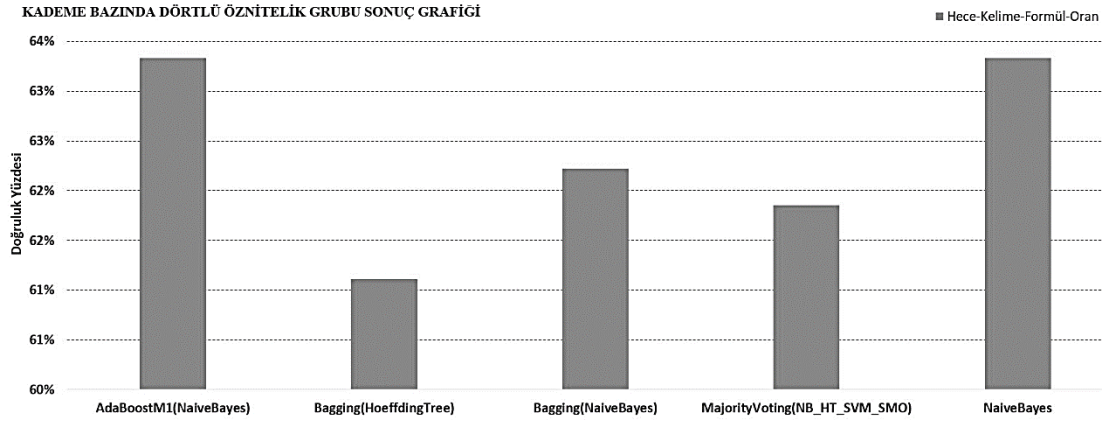
Kademe bazlı sınıflandırma sonucuna göre en iyi başarı oranı MajorityVoting (NB_HT_SVM_SMO) algoritması ile Hece-Kelime-Formül özniteliği için %71.5 olarak elde edilmiştir.

3.3.5. Dörtlü Öznitelik Grubu Sonuçları

Bu deneyde Hece-Kelime-Formül-Oran özniteliği veri seti ile deney yapılmıştır. Deneyde tek öznitelik kullanıldığından, seçilen en iyi beş algoritmanın birbiriyle kıyaslanması yapılmıştır.

Deney sonucunda çıkan sonuçlar kademe bazlı olarak Şekil 17’de verilmiştir.

Kademe bazlı yapılan deneyde, Hece-Kelime-Formül-Oran özniteliği için sınıflandırma sonucuna göre başarı oranı, AdaBoostM1 (NaiveBayes) ve NaiveBayes algoritmaları ile %63.3 olarak elde edilmiştir.

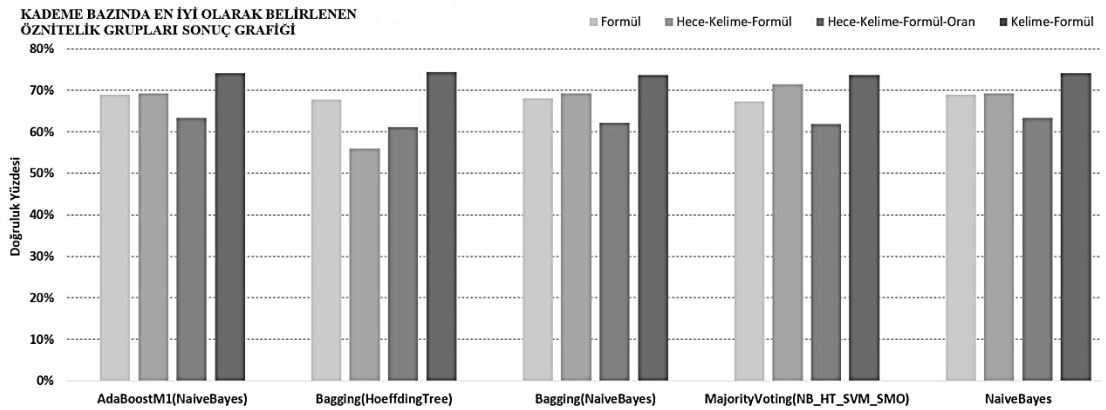


Şekil 17. Kademe bazında dörtlü öznetelik grubu sonuç grafiği

3.3.6. Kademe Bazlı Sonuçların Değerlendirilmesi

Kademe bazında birli, ikili, üçlü ve dörtlü öznetelik grupları kullanılarak analizler yapılmıştır. Çıkan sonuçlara göre en iyi kombinasyonlar Formül, Kelime-Formül, Hece-Kelime-Formül ve Hece-Kelime-Formül-Oran öznetelik gruplarıdır. Bu öznetelik grupları kıyaslandığında en başarılı sonuçların Kelime-Formül (%74.4) özneteliği olduğu görülmüştür. Diğer öznetelikler başarı oranlarına göre sıralandığında; Hece-Kelime-Formül öznetelik grubu (%71.5) ikinci sırada, Formül özneteliği (% 68.9) üçüncü sırada ve Hece-Kelime-Formül-Oran özneteliği (%63.3) ise dördüncü sırada yer almıştır.

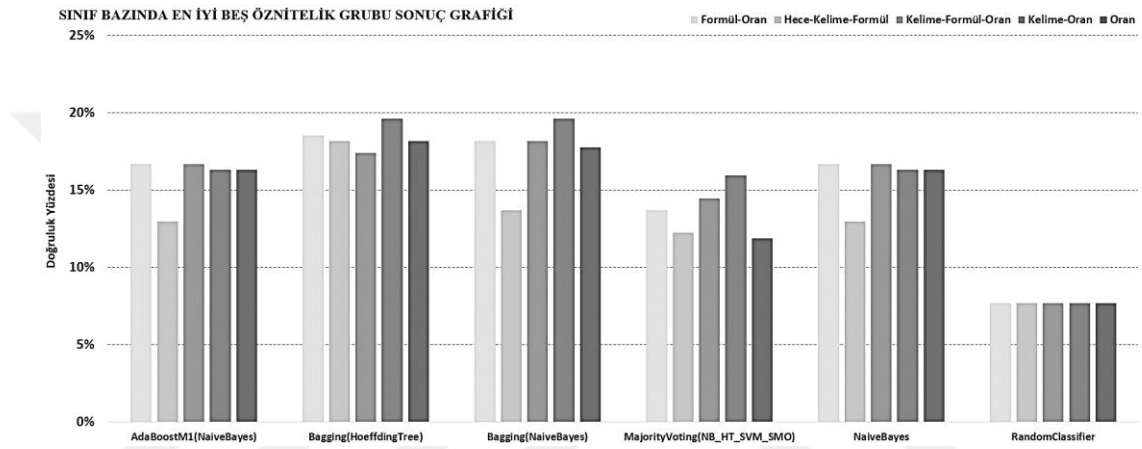
Kademe bazlı yapılan deneylerde en iyi öznetelik grupları kıyaslama grafiği Şekil 18’de gösterilmiştir.



Şekil 18. Kademe bazında en iyi olarak belirlenen öznetelik grupları sonuç grafiği

3.4. SINIF BAZLI SONUÇLAR

Ek1’de verilen tabloya göre sınıf bazlı sonuçlar incelendiğinde en iyi sonucu veren özniteliklerden ilk beş tanesi seçilerek bu deney yapılmıştır. Bu öznitelikler: Formül-Oran, Hece-Kelime-Formül, Kelime-Formül-Oran, Kelime-Oran ve Oran öznitelikleridir. Deney sonucunda çıkan sonuçlar sınıf bazlı olarak Şekil 19’da verilmiştir.



Şekil 19. Sınıf bazında en iyi beş öznitelik grubu sonuç grafiği

Sınıf bazlı yapılan deneyde, Kelime-Oran özniteliği için sınıflandırma sonucuna göre başarı oranı, Bagging (HoeffdingTree) ve Bagging (NaiveBayes) algoritmaları ile %19.6 doğruluk olarak elde etmiştir. Bu sonuçlar sınıf bazlı yapılan deneylerde elde edilen sonuçların en iyisidir. Ayrıca sınıf bazında rastgele sınıflandırma yapıldığında başarı oranının %7.7 olduğu görülmüştür. Diğer öznitelik grupları ile yapılan deneylerin sonuçları %19.6’dan daha küçük olduğu için ayrı ayrı burada verilmemiştir. Detaylı sonuçlar Ek 1’de verilmiştir. Bu sonuçlar okunabilirlik seviyeleri arasında sınıf seviyesinde bir sınıflandırma yapmanın zor olduğunu ortaya koymaktadır.

3.5. TESPİT EDİLEN EN İYİ ÖZİNTELİKLER İLE VARSAYIMLI SONUÇLAR

Eğitim sistemleri başarı kriterine göre değerlendirilerek öğrencinin başarılı veya başarısız olduğuna karar verilir. Bir seviyeyi başarı ile bitiren öğrenci, üst

seviyeye geçmeye hak kazanır. Bu nedenle eğitim sistemlerindeki seviyelerde üst seviyeler alt seviyeleri kapsayacak şekildedir. Yani öğrencilere verilen eğitsel içerikler, kendi seviyesi ve daha alt seviyelerde olmak kaydıyla öğrencinin seviyesine uygun olarak kabul edilebilir.

Bu varsayımla eğitsel içerikler açısından bakıldığında, gerçekte ilkokul seviyesinde olan bir metnin sınıflandırıcı tarafından ortaokul seviyesinde nitelendirilmesi ile bu metnin ilkokul yerine ortaokul öğrencisine önerilmesi, ortaokul seviyesi ilkokul seviyesini kapsayacağından hatalı bir önerme sayılmaz.

Kademe bazında en iyi başarı oranı elde edilen Kelime-Formül özniteliği bu bakış açısıyla değerlendirildiğinde elde edilen başarı oranı %88.9 olmaktadır. Sınıf bazında en iyi başarı oranı elde edilen Kelime-Oran özniteliği bu bakış açısıyla değerlendirildiğinde elde edilen başarı oranı %47.4 olmaktadır.

Bu varsayıma göre kademe bazında gerçekte ilkokul seviyesi, sınıf bazında 1. sınıf seviyesi olan metinler sınıflandırıcı tarafından ortaokul veya lise seviyesi olarak sınıflandırıldığında hata kabul edilmemektedir. Bu nedenle Kademe bazında ilkokul seviyesi, sınıf bazında 1. sınıf seviyesi verileri çıkarılarak Kelime-Formül ve Kelime-Oran öznitelikleri ile yeniden test edildiğinde kademe bazlı başarı oranı %83.3, sınıf bazlı başarı oranı %42.5 olarak gözlemlenmiştir.

SONUÇ

Sonuçlar

Türkçe eğitsel içerikli verilerin okunabilirlik seviyelerine göre sınıflandırılması amacıyla yapılan bu çalışmada yapılan deneyler, Deneysel Sonuçlar bölümünde ele alınmıştır. Bu deneylere göre ulaşılan sonuçlar şu şekildedir:

- Literatürde önerilen okunabilirlik formüllerinin K-12 seviyesinde Türkçe eğitimsel içeriklerinin okunabilirliklerini sınıflandırmadaki başarıları düzeyleri; Ateşman okunabilirlik formülü için %67.4, Çetinkaya-Uzun okunabilirlik formülü için %66.7, ARI okunabilirlik formülü için %65.9 seviyesindedir.
- Eğitsel içeriklerin sınıflandırılmasında başarıyı artırabilecek yeni/özgün özniteliklerin, yapılan deneylerde bulunabildiği gösterilmiştir. Nitekim yapılan deneylerde alınan sonuçlardan Formül özniteliği %68.9, Kelime-Formül özniteliği için %74.4, Hece-Kelime-Formül özniteliği için %71.5 seviyesindedir. Yeni/özgün öznitelik olarak kullanılan Kelime ve Hece öznitelik gruplarının sadece okunabilirlik formülleri ile yapılan sınıflandırmaya göre daha başarılı olduğu tespit edilmiştir. En iyi öznitelik grubu Kelime-Formül özniteliğidir. Performansa etkisi, ayrı ayrı kullanılan okunabilirlik formüllerine göre %7 seviyesinde iken bu tez kapsamında kullandığımız Formül öznitelik grubuna göre %5.5 seviyesindedir.
- Deney sonuçlarına göre en iyi başarı oranı %74 civarındadır. Bu orana sahip algoritmaların başarı sıralaması:
 - 1. sırada Bagging algoritması ile kullanılan HoeffdingTree,
 - 2. sırada AdaBoost algoritması ile kullanılan NaiveBayes,
 - 3. sırada NaiveBayes,

- 4. sırada Bagging algoritması ile kullanılan NaiveBayes,
- 5. sırada NaiveBayes, HoeffdingTree, SVM, SMO algoritmaları ile yapılan MajorityVoting algoritması şeklindedir.

Öneriler

Yapılan deneyler sonucunda kelime listelerinin önemi ortaya çıkmıştır. Bu nedenle kelime listelerinin her sınıf seviyesine göre daha detaylı hazırlanmasının sınıflandırma başarısına katkı sağlayacağı düşünülmektedir. Kelime listelerini ve Türkçe'nin cümle yapısını baz alan yeni bir formül geliştirilebilir. Çalışma farklı derslerle tekrar edilerek veri evreni genişletilebilir.

Yapılan deneylerde en iyi sonuç veren Kelime-Formül özniteliğinde Bagging (HoeffdingTree) algoritmasının hatalı verileri incelendiğinde cümle sayısı açısından çok kısa konularda çok hata yapıldığı görülmektedir. Dört cümle ve daha az sayılı cümleden oluşan konular çıkarıldığında sonuçlarda %3'lük artış görülmektedir.

KAYNAKÇA

- AKTAŞ, Ö. (2006). Türkçe için verimli bir cümle sonu belirleme yöntemi. Akademik Bilişim 2006, BilgiTek IV.
- ATEŞMAN, E. (1997). Türkçede okunabilirliğin ölçülmesi. Ankara Üniversitesi TÖMER Dil Dergisi, (58).71-74.
- BEZİRCİ, B. & YILMAZ, A.E. (2010). Metinlerin okunabilirliğin ölçülmesi üzerine bir yazılım kütüphanesi ve Türkçe için yeni bir okunabilirlik ölçütü. DEÜ Fen Bilimleri Dergisi, 12 (3), 49-62.
- BİLAL D. (2013). Comparing Google's readability of search results to the Flesch readability formulae: A preliminary analysis on children's search queries. Proceedings of 76th ASIS&T Annual Meeting, pp. 1-9.
- CHALL, J. S. (1996). Varying approaches to readability measurement. Revue quebecoise de linguistique, 25, 23-40.
- ÇETİNKAYA, G. (2010). Türkçe metinlerin okunabilirlik düzeylerinin tanımlanması ve sınıflandırılması. Doktora Tezi, Ankara Üniversitesi, Sosyal Bilimler Enstitüsü.
- ÇİFTÇİ, Ö., ÇEÇEN, M. A, MELANLIOĞLU, D. (2007). Altıncı sınıf Türkçe ders kitaplarındaki metinlerin okunabilirlik açısından değerlendirilmesi. Elektronik Sosyal Bilimler Dergisi; 6(22):206-219
- ÇOBAN A. (2014). Okunabilirlik kavramına yönelik bir derleme çalışması. Dil ve Edebiyat Eğitim Dergisi, 9, 96-111.
- DALE, E. & CHALL, J. S. (1948). A formula for predicting readability. Educational research bulletin Jan. 21 and Feb 17, 27:1-20, 37-54.
- DİNÇER B. T. (2004). Türkçe için istatistiksel bir bilgi geri-getirim sistemi. Doktora Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü.
- DUBAY, W. H. (2004). The principles of readability. Alındığı tarih: 15.12.2013, adres: <http://www.impactinformation.com>
- ERASLAN, F. (2008). Eğitsel içerikli web sitelerinin okunabilirlik açısından incelenmesi. Yüksek Lisans Tezi, Gazi Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.

- FLESCHE, R. (1948). A new readability yardstick. *Journal of Applied Psychology* 32: 221-233.
- FRY, E. (1968). A readability formula that saves time. *Journal of Reading*, 11, 513-516, 575-578.
- GOLDBORD, R. (2001). Readable writing by scientists and researchers. *Journal of Environmental Health*, 63, 8, 40-42
- GUNNING, R. (1952). *The technique of clear writing*. McGraw-Hill.
- GÜNEŞ, F. (2000). Çocuk kitaplarının okunabilirlik ölçütleri açısından incelenmesi. I. Ulusal Çocuk Kitapları Sempozyumu. A.Ü. Tömer Dil Öğretim Merkezi.
- HALL M., FRANK E., HOLMES G., PFAHRINGER B., REUTEMANN P., WITTEN I.H. (2009). The Weka data mining software: An update. *SIGKDD Explorations*, 11, 1.
- HEUS, M. & HIEMSTRA, D. (2013). Readability of the web: A study on 1 billion web pages. In: 13th Dutch-Belgian Information Retrieval Workshop, DIR 2013, pp. 66-67.
- KEKLİK, S. (2010). Türkçede 0-6 yaş çocuklarına öğretilmesi gereken, en sık kullanılan 1200 kelime. *Türkiye sosyal araştırmalar dergisi*, 3, 7.
- KEKLİK, S. (2011). Türkçede 7-11 yaş çocuklarına öğretilmesi gereken, en sık kullanılan 1800 kelime. *Türkiye sosyal araştırmalar dergisi*, 2, 2.
- KINCAID, J. P., FISHBURNE, R. P., ROGERS, R. L., CHISSOM B. S. (1975). Derivation of new readability formulas (Automated readability index, Fog count, and Flesch reading ease formula) for navy enlisted personnel. *Research Branch Report*, 8-75.
- KLARE, G. R. (1963). *The measurement of readability*. Iowa State University.
- KÖSE, E. A., (2009). Biyoloji 9 ders kitabında hücre ile ilgili metinlerin okunabilirlik düzeyleri. *Journal of Arts and Sciences*, 12, 141-150.
- MEB (Milli Eğitim Bakanlığı). (t.y.). 2014-2015 yılı ilk ve orta öğretim ders kitapları. Alındığı tarih: 01.07.2015, adres: <http://www.meb.gov.tr/2014-2015-egitim-ogretim-yilinda-okutulacak-ilk-ve-orta-ogretim-ders-kitaplari/duyuru/7013>
- MEB (Milli Eğitim Bakanlığı). (t.y.). Talim ve Terbiye Kurulu Başkanlığı. "Taslak ders kitaplarının incelenmesinde, değerlendirmeye esas olacak kriterler" konulu

- çalışması. Alındığı tarih: 01.07.2015, adres: http://e-mufredat.meb.gov.tr/Dokumanlar/14062748_inceleme_kriterleri_14012013.pdf
- RICHARDS, J. C., PLATT, J., & PLATT, H. (1992). Longman dictionary of language teaching and applied linguistics. London: Longman.
- SENDER R. J., SMITH E. A. (1967). Automated readability index technical report. DTIC Document.
- SÖNMEZ, S. (2003). Kitap tipografisinde okunabilirlik ve görsel kaliteyi etkileyen faktörlerin incelenmesi. Marmara Üniversitesi Fen Bilimleri Enstitüsü Matbaa Eğitimi Anabilim Dalı, Yüksek Lisans Tezi, İstanbul.
- STONE, G. (1996). Readability formulas take bad raps. Alındığı tarih: 09.12.2013, adres: <http://taa.winona.msus.edu/taa/COLUMNS/stone/formulas.html>
- TEMUR, T. (2003). Okunabilirlik (readability) kavramı. TÜBAR, Türkçenin Öğretimi Özel Sayısı,13, 169-180.
- THORNDIKE, E. L. (1921). The teacher's word book. New York: Bureau of Publications. Teachers College, Columbia University.
- TDK (Türk Dil Kurumu). (t.y.). Kısaltmalar dizini. Alındığı tarih: 28.08.2015, adres: http://tdk.gov.tr/index.php?option=com_content&view=article&id=198:Kisaltmalar-Dizini&catid=50:yazm-kurallar&Itemid=132
- TÜİK (Türkiye İstatistik Kurumu). (2015). Bilgi toplumu istatistikleri. Alındığı tarih: 28.08.2015, adres: http://www.tuik.gov.tr/PreIstatistikTablo.do?istab_id=1615
- ZEMBEREK. (t.y.). Açık kaynak kodlu Türkçe doğal dil işleme kütüphanesi. Wikipedia. Alındığı tarih: 01.07.2015, adres: <https://code.google.com/p/zemberek/>
- ZORBAZ, K. Z. (2007). Türkçe ders kitaplarındaki masalların kelime-cümle uzunlukları ve okunabilirlik düzeyleri üzerine bir değerlendirme. Eğitimde Kuram ve Uygulama. 3, 1. 87-101.

EKLER

EK 1. Tüm öznitelik grupları için 22 farklı algoritma ile yapılan test sonuçları

Öznitelik Tipi	Algoritma Tipi	Sınıf Tipi	Başarı
Kelime-Formül	Bagging(HoeffdingTree)	Kademe	%74.44
Kelime-Formül	AdaBoostM1(NaiveBayes)	Kademe	%74.07
Kelime-Formül	NaiveBayes	Kademe	%74.07
Kelime-Formül	Bagging(NaiveBayes)	Kademe	%73.70
Kelime-Formül	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%73.70
Hece-Kelime-Formül	LibSVM	Kademe	%72.96
Kelime-Formül	AdaBoostM1(HoeffdingTree)	Kademe	%72.96
Kelime-Formül	AdaBoostM1(SMO)	Kademe	%72.96
Kelime-Formül	HoeffdingTree	Kademe	%72.96
Kelime-Formül	LibSVM	Kademe	%72.96
Kelime-Formül	SMO	Kademe	%72.96
Hece-Kelime-Formül	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%71.48
Kelime-Formül	Bagging(SMO)	Kademe	%70.37
Kelime-Formül	Stacking(NB_HT_SVM_SMO)	Kademe	%70.37
Kelime-Formül-Oran	Bagging(HoeffdingTree)	Kademe	%69.63
Hece-Kelime-Formül	AdaBoostM1(NaiveBayes)	Kademe	%69.26
Hece-Kelime-Formül	Bagging(NaiveBayes)	Kademe	%69.26
Hece-Kelime-Formül	NaiveBayes	Kademe	%69.26
Formül	AdaBoostM1(NaiveBayes)	Kademe	%68.89
Formül	NaiveBayes	Kademe	%68.89
Kelime	Bagging(SMO)	Kademe	%68.52
Kelime-Formül-Oran	Bagging(NaiveBayes)	Kademe	%68.52
Formül	Bagging(NaiveBayes)	Kademe	%68.15
Formül	LibSVM	Kademe	%68.15
Formül	Bagging(HoeffdingTree)	Kademe	%67.78
Formül	Stacking(NB_HT_SVM_SMO)	Kademe	%67.78
Ateşman	Bagging(NaiveBayes)	Kademe	%67.41
Formül	AdaBoostM1(SMO)	Kademe	%67.41
Formül	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%67.41
Formül	SMO	Kademe	%67.41
Kelime	Bagging(NaiveBayes)	Kademe	%67.41
Kelime	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%67.41
Hece-Kelime-Formül	AdaBoostM1(SMO)	Kademe	%67.04
Hece-Kelime-Formül	SMO	Kademe	%67.04
Kelime	AdaBoostM1(SMO)	Kademe	%67.04
Kelime	Bagging(HoeffdingTree)	Kademe	%67.04
Kelime	SMO	Kademe	%67.04
Kelime-Formül-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%67.04
Çetinkaya-Uzun	AdaBoostM1(NaiveBayes)	Kademe	%66.67
Çetinkaya-Uzun	NaiveBayes	Kademe	%66.67
Kelime	LibSVM	Kademe	%66.67
Ateşman	Bagging(HoeffdingTree)	Kademe	%66.30
Kelime	Stacking(NB_HT_SVM_SMO)	Kademe	%66.30
Kelime-Formül-Oran	AdaBoostM1(NaiveBayes)	Kademe	%66.30
Kelime-Formül-Oran	NaiveBayes	Kademe	%66.30
Ari	AdaBoostM1(NaiveBayes)	Kademe	%65.93
Ari	NaiveBayes	Kademe	%65.93
Ateşman	AdaBoostM1(NaiveBayes)	Kademe	%65.93
Ateşman	NaiveBayes	Kademe	%65.93

Çetinkaya-Uzun	Bagging(NaiveBayes)	Kademe	%65.93
Kelime	AdaBoostM1(NaiveBayes)	Kademe	%65.93
Kelime	NaiveBayes	Kademe	%65.93
Ari	Bagging(NaiveBayes)	Kademe	%65.56
Hece-Kelime-Formül	Bagging(SMO)	Kademe	%65.56
Kelime-Formül	AttributeSelectedClassifier	Kademe	%65.56
Kelime-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%65.56
Çetinkaya-Uzun	Bagging(HoeffdingTree)	Kademe	%65.19
Kelime-Formül-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%65.19
Kelime-Formül-Oran	HoeffdingTree	Kademe	%65.19
Kelime-Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%65.19
Çetinkaya-Uzun	LibSVM	Kademe	%64.81
Çetinkaya-Uzun	Stacking(NB_HT_SVM_SMO)	Kademe	%64.81
Hece-Kelime	AdaBoostM1(SMO)	Kademe	%64.81
Hece-Kelime	SMO	Kademe	%64.81
Hece-Kelime-Formül-Oran	AdaBoostM1(SMO)	Kademe	%64.81
Hece-Kelime-Formül-Oran	SMO	Kademe	%64.81
Kelime	AttributeSelectedClassifier	Kademe	%64.81
Kelime-Formül-Oran	Bagging(SMO)	Kademe	%64.81
Kelime-Oran	Bagging(NaiveBayes)	Kademe	%64.81
Kelime-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%64.81
Formül	Bagging(SMO)	Kademe	%64.44
Kelime-Oran	Bagging(HoeffdingTree)	Kademe	%64.44
Çetinkaya-Uzun	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%64.07
Çetinkaya-Uzun	SimpleLogistic	Kademe	%64.07
Hece-Kelime-Formül-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%64.07
Kelime	LogitBoost	Kademe	%64.07
Ateşman	AdaBoostM1(SMO)	Kademe	%63.70
Ateşman	SMO	Kademe	%63.70
Çetinkaya-Uzun	AdaBoostM1(SMO)	Kademe	%63.70
Çetinkaya-Uzun	SMO	Kademe	%63.70
Hece-Kelime-Formül	Stacking(NB_HT_SVM_SMO)	Kademe	%63.70
Kelime	ClassificationViaRegression	Kademe	%63.70
Kelime	SimpleLogistic	Kademe	%63.70
Kelime-Oran	AdaBoostM1(SMO)	Kademe	%63.70
Kelime-Oran	SMO	Kademe	%63.70
Ateşman	LibSVM	Kademe	%63.33
Ateşman	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%63.33
Ateşman	SimpleLogistic	Kademe	%63.33
Hece-Kelime-Formül-Oran	AdaBoostM1(NaiveBayes)	Kademe	%63.33
Hece-Kelime-Formül-Oran	NaiveBayes	Kademe	%63.33
Hece-Kelime-Oran	AdaBoostM1(SMO)	Kademe	%63.33
Hece-Kelime-Oran	SMO	Kademe	%63.33
Kelime-Formül	Bagging(Default)	Kademe	%63.33
Kelime-Oran	AdaBoostM1(NaiveBayes)	Kademe	%63.33
Kelime-Oran	ClassificationViaRegression	Kademe	%63.33
Kelime-Oran	NaiveBayes	Kademe	%63.33
Kelime-Formül-Oran	AdaBoostM1(SMO)	Kademe	%62.96
Kelime-Formül-Oran	ClassificationViaRegression	Kademe	%62.96
Kelime-Formül-Oran	SMO	Kademe	%62.96
Ari	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%62.59
Ateşman	MultilayerPerceptron	Kademe	%62.59
Formül-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%62.59
Formül-Oran	HoeffdingTree	Kademe	%62.59
Kelime	Bagging(Default)	Kademe	%62.59
Kelime-Formül	SimpleLogistic	Kademe	%62.59
Kelime-Formül-Oran	LibSVM	Kademe	%62.59
Kelime-Oran	Bagging(SMO)	Kademe	%62.59
Ari	AdaBoostM1(SMO)	Kademe	%62.22

Ari	LibSVM	Kademe	%62.22
Ari	SMO	Kademe	%62.22
Ari	Stacking(NB_HT_SVM_SMO)	Kademe	%62.22
Ateşman	ClassificationViaRegression	Kademe	%62.22
Formül-Oran	Bagging(HoeffdingTree)	Kademe	%62.22
Hece-Kelime-Formül-Oran	Bagging(NaiveBayes)	Kademe	%62.22
Ari	Bagging(Default)	Kademe	%61.85
Formül-Oran	AdaBoostM1(NaiveBayes)	Kademe	%61.85
Formül-Oran	NaiveBayes	Kademe	%61.85
Hece-Kelime-Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%61.85
Kelime-Formül	J48	Kademe	%61.85
Ari	Bagging(HoeffdingTree)	Kademe	%61.48
Ateşman	Bagging(SMO)	Kademe	%61.48
Çetinkaya-Uzun	LogitBoost	Kademe	%61.11
Çetinkaya-Uzun	MultilayerPerceptron	Kademe	%61.11
Hece-Kelime	Bagging(SMO)	Kademe	%61.11
Hece-Kelime-Formül-Oran	Bagging(HoeffdingTree)	Kademe	%61.11
Kelime-Formül	ClassificationViaRegression	Kademe	%61.11
Kelime-Formül	LogitBoost	Kademe	%61.11
Kelime-Formül-Oran	SimpleLogistic	Kademe	%61.11
Kelime-Oran	SimpleLogistic	Kademe	%61.11
Ari	LogitBoost	Kademe	%60.74
Ateşman	AttributeSelectedClassifier	Kademe	%60.74
Ateşman	J48	Kademe	%60.74
Formül-Oran	Bagging(NaiveBayes)	Kademe	%60.74
Ateşman	Stacking(NB_HT_SVM_SMO)	Kademe	%60.37
Çetinkaya-Uzun	Bagging(SMO)	Kademe	%60.37
Hece-Kelime-Oran	SimpleLogistic	Kademe	%60.37
Kelime-Formül	MultilayerPerceptron	Kademe	%60.37
Kelime-Formül	RandomForest	Kademe	%60.37
Ari	Bagging(SMO)	Kademe	%60.00
Ari	ClassificationViaRegression	Kademe	%60.00
Ari	SimpleLogistic	Kademe	%60.00
Hece-Kelime-Formül-Oran	SimpleLogistic	Kademe	%60.00
Kelime	J48	Kademe	%60.00
Kelime-Oran	LibSVM	Kademe	%60.00
Çetinkaya-Uzun	ClassificationViaRegression	Kademe	%59.63
Formül-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%59.63
Hece-Kelime-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%59.63
Hece-Kelime	SimpleLogistic	Kademe	%59.26
Hece-Kelime-Formül	ClassificationViaRegression	Kademe	%59.26
Formül	ClassificationViaRegression	Kademe	%58.89
Hece-Kelime	ClassificationViaRegression	Kademe	%58.89
Hece-Kelime-Formül	FilteredClassifier	Kademe	%58.89
Hece-Kelime-Formül-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%58.89
Hece-Kelime-Formül-Oran	AttributeSelectedClassifier	Kademe	%58.89
Hece-Kelime-Formül-Oran	Bagging(SMO)	Kademe	%58.89
Hece-Kelime-Formül-Oran	HoeffdingTree	Kademe	%58.89
Hece-Kelime-Oran	AttributeSelectedClassifier	Kademe	%58.89
Hece-Kelime-Oran	Bagging(NaiveBayes)	Kademe	%58.89
Hece-Kelime-Oran	Bagging(SMO)	Kademe	%58.89
Kelime-Formül	FilteredClassifier	Kademe	%58.89
Kelime-Formül-Oran	AttributeSelectedClassifier	Kademe	%58.89
Kelime-Oran	AttributeSelectedClassifier	Kademe	%58.89
Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%58.52
Hece-Kelime	RandomForest	Kademe	%58.52
Kelime	MultilayerPerceptron	Kademe	%58.52
Formül-Oran	FilteredClassifier	Kademe	%58.15
Hece-Formül-Oran	FilteredClassifier	Kademe	%58.15

Hece-Oran	FilteredClassifier	Kademe	%58.15
Oran	FilteredClassifier	Kademe	%58.15
Çetinkaya-Uzun	AttributeSelecedClassifier	Kademe	%57.78
Çetinkaya-Uzun	J48	Kademe	%57.78
Hece-Oran	AttributeSelecedClassifier	Kademe	%57.78
Oran	AttributeSelecedClassifier	Kademe	%57.78
Oran	Bagging(HoeffdingTree)	Kademe	%57.78
Formül-Oran	AttributeSelecedClassifier	Kademe	%57.41
Hece-Formül-Oran	AttributeSelecedClassifier	Kademe	%57.41
Hece-Kelime-Formül	SimpleLogistic	Kademe	%57.41
Hece-Kelime-Formül-Oran	ClassificationViaRegression	Kademe	%57.41
Kelime-Oran	MultilayerPerceptron	Kademe	%57.41
Oran	AdaBoostM1(HoeffdingTree)	Kademe	%57.41
Oran	Bagging(NaiveBayes)	Kademe	%57.41
Oran	HoeffdingTree	Kademe	%57.41
Ari	MultilayerPerceptron	Kademe	%57.04
Ari	RandomForest	Kademe	%57.04
Hece-Kelime-Oran	Bagging(HoeffdingTree)	Kademe	%57.04
Hece-Kelime-Oran	FilteredClassifier	Kademe	%57.04
Hece-Kelime-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%57.04
Kelime	RandomForest	Kademe	%57.04
Kelime-Oran	FilteredClassifier	Kademe	%57.04
Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%57.04
Hece-Formül-Oran	AdaBoostM1(NaiveBayes)	Kademe	%56.67
Hece-Formül-Oran	NaiveBayes	Kademe	%56.67
Hece-Kelime-Formül	AdaBoostM1(HoeffdingTree)	Kademe	%56.67
Hece-Kelime-Formül	HoeffdingTree	Kademe	%56.67
Hece-Kelime-Formül-Oran	FilteredClassifier	Kademe	%56.67
Hece-Kelime-Formül-Oran	LibSVM	Kademe	%56.67
Hece-Kelime-Oran	AdaBoostM1(NaiveBayes)	Kademe	%56.67
Hece-Kelime-Oran	NaiveBayes	Kademe	%56.67
Kelime-Formül-Oran	FilteredClassifier	Kademe	%56.67
Oran	AdaBoostM1(NaiveBayes)	Kademe	%56.67
Oran	NaiveBayes	Kademe	%56.67
Hece-Formül	AdaBoostM1(NaiveBayes)	Kademe	%56.30
Hece-Formül	NaiveBayes	Kademe	%56.30
Hece-Formül-Oran	Bagging(NaiveBayes)	Kademe	%56.30
Hece-Kelime-Oran	ClassificationViaRegression	Kademe	%56.30
Ateşman	Bagging(Default)	Kademe	%55.93
Hece-Kelime-Formül	Bagging(HoeffdingTree)	Kademe	%55.93
Hece-Kelime	Stacking(NB_HT_SVM_SMO)	Kademe	%55.19
Kelime-Formül-Oran	MultilayerPerceptron	Kademe	%55.19
Formül-Oran	Bagging(SMO)	Kademe	%54.81
Hece-Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%54.81
Hece-Formül-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%54.81
Hece-Kelime-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%54.81
Hece-Kelime-Oran	HoeffdingTree	Kademe	%54.81
Hece-Kelime-Oran	LibSVM	Kademe	%54.81
Kelime-Oran	Bagging(Default)	Kademe	%54.81
Formül	LogitBoost	Kademe	%54.44
Hece-Formül	AdaBoostM1(HoeffdingTree)	Kademe	%54.44
Hece-Formül	Bagging(NaiveBayes)	Kademe	%54.44
Hece-Formül	HoeffdingTree	Kademe	%54.44
Hece-Kelime	Bagging(Default)	Kademe	%54.44
Hece-Kelime-Formül	LogitBoost	Kademe	%54.44
Ateşman	LogitBoost	Kademe	%54.07
Hece-Formül	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%54.07
Hece-Formül-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%54.07
Hece-Formül-Oran	HoeffdingTree	Kademe	%54.07

Formül-Oran	AdaBoostM1(SMO)	Kademe	%53.70
Formül-Oran	SMO	Kademe	%53.70
Kelime-Formül-Oran	LogitBoost	Kademe	%53.70
Hece-Kelime	AdaBoostM1(Default)	Kademe	%53.33
Hece-Kelime	FilteredClassifier	Kademe	%53.33
Hece-Kelime-Formül	AdaBoostM1(Default)	Kademe	%53.33
Kelime	AdaBoostM1(Default)	Kademe	%53.33
Kelime	FilteredClassifier	Kademe	%53.33
Kelime-Formül	AdaBoostM1(Default)	Kademe	%53.33
Kelime-Oran	LogitBoost	Kademe	%53.33
Formül	Bagging(Default)	Kademe	%52.96
Hece-Formül-Oran	Bagging(HoeffdingTree)	Kademe	%52.96
Hece-Kelime	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%52.96
Çetinkaya-Uzun	AdaBoostM1(Default)	Kademe	%52.59
Çetinkaya-Uzun	FilteredClassifier	Kademe	%52.59
Hece-Kelime-Formül-Oran	RandomForest	Kademe	%52.59
Oran	J48	Kademe	%52.59
Formül	AttributeSelecedetClassifier	Kademe	%52.22
Formül	J48	Kademe	%52.22
Formül-Oran	LibSVM	Kademe	%52.22
Hece-Kelime	MultilayerPerceptron	Kademe	%52.22
Kelime-Oran	RandomForest	Kademe	%52.22
Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%52.22
Çetinkaya-Uzun	Bagging(Default)	Kademe	%51.85
Formül	SimpleLogistic	Kademe	%51.85
Hece-Formül	Stacking(NB_HT_SVM_SMO)	Kademe	%51.85
Hece-Kelime	Bagging(NaiveBayes)	Kademe	%51.85
Hece-Kelime-Formül	MultilayerPerceptron	Kademe	%51.85
Hece-Kelime-Formül-Oran	MultilayerPerceptron	Kademe	%51.85
Formül-Oran	SimpleLogistic	Kademe	%51.48
Hece-Kelime-Formül	RandomForest	Kademe	%51.48
Hece-Kelime-Formül-Oran	Bagging(Default)	Kademe	%51.48
Hece-Kelime-Oran	Bagging(Default)	Kademe	%51.48
Hece-Oran	Bagging(NaiveBayes)	Kademe	%51.48
Ateşman	AdaBoostM1(Default)	Kademe	%51.11
Ateşman	FilteredClassifier	Kademe	%51.11
Formül	AdaBoostM1(Default)	Kademe	%51.11
Formül	FilteredClassifier	Kademe	%51.11
Formül-Oran	J48	Kademe	%51.11
Hece-Formül	AdaBoostM1(Default)	Kademe	%51.11
Hece-Formül	FilteredClassifier	Kademe	%51.11
Hece-Formül	LibSVM	Kademe	%51.11
Hece-Kelime-Formül	AttributeSelecedetClassifier	Kademe	%51.11
Hece-Kelime-Formül	Bagging(Default)	Kademe	%51.11
Hece-Oran	AdaBoostM1(NaiveBayes)	Kademe	%51.11
Hece-Oran	NaiveBayes	Kademe	%51.11
Hece-Oran	Stacking(NB_HT_SVM_SMO)	Kademe	%51.11
Kelime-Formül-Oran	Bagging(Default)	Kademe	%51.11
Kelime-Formül-Oran	RandomForest	Kademe	%51.11
Oran	SimpleLogistic	Kademe	%51.11
Hece-Formül-Oran	LibSVM	Kademe	%50.74
Hece-Kelime	LogitBoost	Kademe	%50.74
Hece-Oran	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%50.37
Formül-Oran	ClassificationViaRegression	Kademe	%50.00
Hece-Kelime	AttributeSelecedetClassifier	Kademe	%50.00
Hece-Kelime-Oran	RandomForest	Kademe	%50.00
Oran	Bagging(SMO)	Kademe	%50.00
Ari	AttributeSelecedetClassifier	Kademe	%49.63
Ari	J48	Kademe	%49.63

Hece-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%49.63
Hece-Oran	HoeffdingTree	Kademe	%49.63
Formül-Oran	MultilayerPerceptron	Kademe	%49.26
Hece-Kelime	LibSVM	Kademe	%49.26
Hece-Oran	AdaBoostM1(SMO)	Kademe	%49.26
Hece-Oran	Bagging(HoeffdingTree)	Kademe	%49.26
Hece-Oran	SMO	Kademe	%49.26
Kelime-Formül-Oran	J48	Kademe	%49.26
Oran	MultilayerPerceptron	Kademe	%49.26
Hece-Formül-Oran	AdaBoostM1(SMO)	Kademe	%48.89
Hece-Formül-Oran	SMO	Kademe	%48.89
Oran	AdaBoostM1(SMO)	Kademe	%48.89
Oran	SMO	Kademe	%48.89
Formül	MultilayerPerceptron	Kademe	%48.52
Hece-Formül	AdaBoostM1(SMO)	Kademe	%48.52
Hece-Formül	SMO	Kademe	%48.52
Hece-Kelime	AdaBoostM1(NaiveBayes)	Kademe	%48.52
Hece-Kelime	NaiveBayes	Kademe	%48.52
Hece-Kelime-Formül-Oran	LogitBoost	Kademe	%48.52
Hece-Kelime-Oran	LogitBoost	Kademe	%48.52
Çetinkaya-Uzun	RandomForest	Kademe	%48.15
Hece-Formül-Oran	Bagging(SMO)	Kademe	%48.15
Hece-Formül	Bagging(HoeffdingTree)	Kademe	%47.78
Hece-Formül-Oran	J48	Kademe	%47.78
Oran	ClassificationViaRegression	Kademe	%47.78
Formül	RandomForest	Kademe	%47.41
Formül-Oran	LogitBoost	Kademe	%47.41
Kelime-Oran	J48	Kademe	%47.41
Hece-Kelime-Oran	MultilayerPerceptron	Kademe	%47.04
Oran	RandomForest	Kademe	%47.04
Hece-Formül	Bagging(Default)	Kademe	%46.67
Hece-Formül	LogitBoost	Kademe	%46.67
Hece-Kelime-Oran	J48	Kademe	%46.67
Formül-Oran	Bagging(Default)	Kademe	%46.30
Hece-Formül	RandomForest	Kademe	%46.30
Hece-Oran	Bagging(SMO)	Kademe	%46.30
Hece-Oran	LibSVM	Kademe	%46.30
Oran	LibSVM	Kademe	%46.30
Oran	LogitBoost	Kademe	%46.30
Formül-Oran	AdaBoostM1(Default)	Kademe	%45.93
Hece-Formül	Bagging(SMO)	Kademe	%45.93
Hece-Formül-Oran	AdaBoostM1(Default)	Kademe	%45.93
Hece-Kelime	J48	Kademe	%45.93
Hece-Kelime-Formül-Oran	AdaBoostM1(Default)	Kademe	%45.93
Hece-Kelime-Oran	AdaBoostM1(Default)	Kademe	%45.93
Hece-Oran	AdaBoostM1(Default)	Kademe	%45.93
Kelime-Formül-Oran	AdaBoostM1(Default)	Kademe	%45.93
Kelime-Oran	AdaBoostM1(Default)	Kademe	%45.93
Oran	AdaBoostM1(Default)	Kademe	%45.93
Hece-Kelime	AdaBoostM1(HoeffdingTree)	Kademe	%45.56
Hece-Kelime	HoeffdingTree	Kademe	%45.56
Hece-Kelime-Formül-Oran	J48	Kademe	%45.19
Hece-Oran	Bagging(Default)	Kademe	%45.19
Ateşman	RandomForest	Kademe	%44.81
Formül-Oran	RandomForest	Kademe	%44.81
Hece-Kelime-Formül	J48	Kademe	%44.81
Hece-Oran	LogitBoost	Kademe	%44.81
Oran	Bagging(Default)	Kademe	%44.07
Hece-Formül	SimpleLogistic	Kademe	%43.33

Hece-Formül-Oran	Bagging(Default)	Kademe	%43.33
Hece-Formül-Oran	RandomForest	Kademe	%43.33
Hece-Formül	AttributeSelecetedClassifier	Kademe	%42.96
Hece-Oran	MultilayerPerceptron	Kademe	%42.96
Hece-Formül-Oran	SimpleLogistic	Kademe	%42.59
Hece-Oran	SimpleLogistic	Kademe	%42.59
Hece-Kelime	Bagging(HoeffdingTree)	Kademe	%41.85
Kelime-Oran	AdaBoostM1(HoeffdingTree)	Kademe	%41.85
Kelime-Oran	HoeffdingTree	Kademe	%41.85
Hece-Formül	ClassificationViaRegression	Kademe	%41.48
Hece-Formül-Oran	LogitBoost	Kademe	%41.48
Hece-Oran	J48	Kademe	%41.48
Hece-Oran	RandomForest	Kademe	%41.48
Hece-Formül-Oran	ClassificationViaRegression	Kademe	%40.00
Hece-Formül-Oran	MultilayerPerceptron	Kademe	%40.00
Hece	MultilayerPerceptron	Kademe	%39.26
Hece-Formül	MultilayerPerceptron	Kademe	%39.26
Ari	AdaBoostM1(Default)	Kademe	%38.52
Ari	FilteredClassifier	Kademe	%38.52
Hece-Formül	J48	Kademe	%38.15
Hece-Oran	ClassificationViaRegression	Kademe	%37.78
Hece	AttributeSelecetedClassifier	Kademe	%36.67
Ari	AdaBoostM1(HoeffdingTree)	Kademe	%36.30
Ari	HoeffdingTree	Kademe	%36.30
Ateşman	AdaBoostM1(HoeffdingTree)	Kademe	%33.33
Ateşman	HoeffdingTree	Kademe	%33.33
Çetinkaya-Uzun	AdaBoostM1(HoeffdingTree)	Kademe	%33.33
Çetinkaya-Uzun	HoeffdingTree	Kademe	%33.33
Formül	AdaBoostM1(HoeffdingTree)	Kademe	%33.33
Formül	HoeffdingTree	Kademe	%33.33
Hece	FilteredClassifier	Kademe	%33.33
Kelime	AdaBoostM1(HoeffdingTree)	Kademe	%32.96
Kelime	HoeffdingTree	Kademe	%32.96
Hece	Stacking(NB_HT_SVM_SMO)	Kademe	%32.59
Hece	RandomForest	Kademe	%30.00
Hece	Bagging(Default)	Kademe	%28.89
Hece	LibSVM	Kademe	%28.89
Hece	ClassificationViaRegression	Kademe	%28.15
Hece	AdaBoostM1(Default)	Kademe	%27.78
Hece	LogitBoost	Kademe	%27.78
Hece	J48	Kademe	%27.04
Hece	AdaBoostM1(HoeffdingTree)	Kademe	%26.67
Hece	Bagging(HoeffdingTree)	Kademe	%26.67
Hece	HoeffdingTree	Kademe	%26.67
Hece	Bagging(NaiveBayes)	Kademe	%24.81
Hece	Bagging(SMO)	Kademe	%24.07
Hece	MajorityVoting(NB_HT_SVM_SMO)	Kademe	%24.07
Hece	AdaBoostM1(NaiveBayes)	Kademe	%23.70
Hece	NaiveBayes	Kademe	%23.70
Hece	AdaBoostM1(SMO)	Kademe	%22.96
Hece	SMO	Kademe	%22.96
Hece	SimpleLogistic	Kademe	%21.11
Kelime-Oran	Bagging(HoeffdingTree)	Sınıf	%19.63
Kelime-Oran	Bagging(NaiveBayes)	Sınıf	%19.63
Kelime-Formül-Oran	SimpleLogistic	Sınıf	%18.89
Formül-Oran	Bagging(HoeffdingTree)	Sınıf	%18.52
Kelime-Oran	SimpleLogistic	Sınıf	%18.52
Formül-Oran	Bagging(NaiveBayes)	Sınıf	%18.15
Hece-Kelime-Formül	Bagging(HoeffdingTree)	Sınıf	%18.15

Kelime-Formül-Oran	Bagging(NaiveBayes)	Sımf	%18.15
Kelime-Oran	AdaBoostM1(HoeffdingTree)	Sımf	%18.15
Kelime-Oran	HoeffdingTree	Sımf	%18.15
Oran	Bagging(HoeffdingTree)	Sımf	%18.15
Hece-Kelime	SimpleLogistic	Sımf	%17.78
Oran	AdaBoostM1(HoeffdingTree)	Sımf	%17.78
Oran	Bagging(NaiveBayes)	Sımf	%17.78
Oran	HoeffdingTree	Sımf	%17.78
Oran	RandomForest	Sımf	%17.78
Hece-Kelime	Bagging(HoeffdingTree)	Sımf	%17.41
Hece-Kelime-Formül-Oran	Bagging(HoeffdingTree)	Sımf	%17.41
Hece-Kelime-Formül-Oran	SimpleLogistic	Sımf	%17.41
Hece-Kelime-Oran	AdaBoostM1(NaiveBayes)	Sımf	%17.41
Hece-Kelime-Oran	NaiveBayes	Sımf	%17.41
Kelime	SimpleLogistic	Sımf	%17.41
Kelime-Formül	SimpleLogistic	Sımf	%17.41
Kelime-Formül-Oran	Bagging(HoeffdingTree)	Sımf	%17.41
Formül	Bagging(HoeffdingTree)	Sımf	%17.04
Hece-Kelime	Bagging(SMO)	Sımf	%17.04
Hece-Kelime-Formül	SimpleLogistic	Sımf	%17.04
Oran	Bagging(Default)	Sımf	%17.04
Formül	Bagging(NaiveBayes)	Sımf	%16.67
Formül-Oran	AdaBoostM1(NaiveBayes)	Sımf	%16.67
Formül-Oran	NaiveBayes	Sımf	%16.67
Hece-Formül-Oran	Bagging(Default)	Sımf	%16.67
Hece-Kelime-Formül-Oran	AdaBoostM1(HoeffdingTree)	Sımf	%16.67
Hece-Kelime-Formül-Oran	HoeffdingTree	Sımf	%16.67
Hece-Oran	Bagging(HoeffdingTree)	Sımf	%16.67
Kelime-Formül-Oran	AdaBoostM1(NaiveBayes)	Sımf	%16.67
Kelime-Formül-Oran	NaiveBayes	Sımf	%16.67
Kelime-Formül-Oran	Stacking(NB_HT_SVM_SMO)	Sımf	%16.67
Hece-Kelime-Formül-Oran	Bagging(Default)	Sımf	%16.30
Hece-Kelime-Formül-Oran	Bagging(NaiveBayes)	Sımf	%16.30
Hece-Kelime-Oran	RandomForest	Sımf	%16.30
Hece-Oran	Bagging(Default)	Sımf	%16.30
Kelime	Bagging(Default)	Sımf	%16.30
Kelime	ClassificationViaRegression	Sımf	%16.30
Kelime-Formül	AdaBoostM1(HoeffdingTree)	Sımf	%16.30
Kelime-Formül	HoeffdingTree	Sımf	%16.30
Kelime-Oran	AdaBoostM1(NaiveBayes)	Sımf	%16.30
Kelime-Oran	Bagging(Default)	Sımf	%16.30
Kelime-Oran	NaiveBayes	Sımf	%16.30
Oran	AdaBoostM1(NaiveBayes)	Sımf	%16.30
Oran	NaiveBayes	Sımf	%16.30
Formül	AdaBoostM1(HoeffdingTree)	Sımf	%15.93
Formül	AdaBoostM1(NaiveBayes)	Sımf	%15.93
Formül	HoeffdingTree	Sımf	%15.93
Formül	NaiveBayes	Sımf	%15.93
Formül-Oran	Bagging(Default)	Sımf	%15.93
Hece-Formül-Oran	RandomForest	Sımf	%15.93
Hece-Kelime-Formül	Bagging(SMO)	Sımf	%15.93
Hece-Kelime-Formül-Oran	Bagging(SMO)	Sımf	%15.93
Hece-Kelime-Formül-Oran	RandomForest	Sımf	%15.93
Hece-Kelime-Oran	AdaBoostM1(HoeffdingTree)	Sımf	%15.93
Hece-Kelime-Oran	Bagging(HoeffdingTree)	Sımf	%15.93
Hece-Kelime-Oran	HoeffdingTree	Sımf	%15.93
Kelime-Formül	Bagging(SMO)	Sımf	%15.93
Kelime-Formül	MajorityVoting(NB_HT_SVM_SMO)	Sımf	%15.93
Kelime-Formül	RandomForest	Sımf	%15.93

Kelime-Formül-Oran	Bagging(Default)	Sınıf	% 15.93
Kelime-Formül-Oran	ClassificationViaRegression	Sınıf	% 15.93
Kelime-Formül-Oran	J48	Sınıf	% 15.93
Kelime-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	% 15.93
Formül-Oran	AdaBoostM1(HoeffdingTree)	Sınıf	% 15.56
Formül-Oran	HoeffdingTree	Sınıf	% 15.56
Formül-Oran	RandomForest	Sınıf	% 15.56
Hece-Kelime	LogitBoost	Sınıf	% 15.56
Hece-Kelime-Oran	AdaBoostM1(SMO)	Sınıf	% 15.56
Hece-Kelime-Oran	SMO	Sınıf	% 15.56
Hece-Oran	RandomForest	Sınıf	% 15.56
Hece-Oran	SimpleLogistic	Sınıf	% 15.56
Kelime	MultilayerPerceptron	Sınıf	% 15.56
Kelime-Formül	AdaBoostM1(NaiveBayes)	Sınıf	% 15.56
Kelime-Formül	NaiveBayes	Sınıf	% 15.56
Kelime-Oran	RandomForest	Sınıf	% 15.56
Formül	Stacking(NB_HT_SVM_SMO)	Sınıf	% 15.19
Hece-Formül	MultilayerPerceptron	Sınıf	% 15.19
Hece-Kelime-Formül	AdaBoostM1(SMO)	Sınıf	% 15.19
Hece-Kelime-Formül	RandomForest	Sınıf	% 15.19
Hece-Kelime-Formül	SMO	Sınıf	% 15.19
Hece-Kelime-Formül-Oran	AdaBoostM1(NaiveBayes)	Sınıf	% 15.19
Hece-Kelime-Formül-Oran	NaiveBayes	Sınıf	% 15.19
Hece-Kelime-Oran	MultilayerPerceptron	Sınıf	% 15.19
Kelime	LogitBoost	Sınıf	% 15.19
Kelime-Formül	Bagging(Default)	Sınıf	% 15.19
Kelime-Formül	Bagging(HoeffdingTree)	Sınıf	% 15.19
Kelime-Formül	Bagging(NaiveBayes)	Sınıf	% 15.19
Kelime-Formül	ClassificationViaRegression	Sınıf	% 15.19
Kelime-Formül-Oran	AdaBoostM1(HoeffdingTree)	Sınıf	% 15.19
Kelime-Formül-Oran	HoeffdingTree	Sınıf	% 15.19
Kelime-Oran	ClassificationViaRegression	Sınıf	% 15.19
Ari	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.81
Ateşman	Bagging(HoeffdingTree)	Sınıf	% 14.81
Çetinkaya-Uzun	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.81
Hece-Formül	ClassificationViaRegression	Sınıf	% 14.81
Hece-Kelime	AdaBoostM1(HoeffdingTree)	Sınıf	% 14.81
Hece-Kelime	ClassificationViaRegression	Sınıf	% 14.81
Hece-Kelime	HoeffdingTree	Sınıf	% 14.81
Hece-Kelime-Formül	ClassificationViaRegression	Sınıf	% 14.81
Hece-Kelime-Formül	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.81
Hece-Kelime-Formül-Oran	MultilayerPerceptron	Sınıf	% 14.81
Hece-Kelime-Oran	Bagging(Default)	Sınıf	% 14.81
Hece-Kelime-Oran	Bagging(SMO)	Sınıf	% 14.81
Hece-Kelime-Oran	SimpleLogistic	Sınıf	% 14.81
Hece-Oran	AdaBoostM1(HoeffdingTree)	Sınıf	% 14.81
Hece-Oran	HoeffdingTree	Sınıf	% 14.81
Kelime-Formül	MultilayerPerceptron	Sınıf	% 14.81
Kelime-Formül-Oran	RandomForest	Sınıf	% 14.81
Formül-Oran	AttributeSelectedClassifier	Sınıf	% 14.44
Formül-Oran	SimpleLogistic	Sınıf	% 14.44
Hece-Formül	SimpleLogistic	Sınıf	% 14.44
Hece-Formül-Oran	AttributeSelectedClassifier	Sınıf	% 14.44
Hece-Kelime	MultilayerPerceptron	Sınıf	% 14.44
Hece-Kelime-Formül	AdaBoostM1(HoeffdingTree)	Sınıf	% 14.44
Hece-Kelime-Formül	HoeffdingTree	Sınıf	% 14.44
Hece-Kelime-Formül-Oran	AdaBoostM1(SMO)	Sınıf	% 14.44
Hece-Kelime-Formül-Oran	SMO	Sınıf	% 14.44
Hece-Kelime-Oran	Bagging(NaiveBayes)	Sınıf	% 14.44

Hece-Kelime-Oran	ClassificationViaRegression	Sınıf	% 14.44
Hece-Kelime-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	% 14.44
Hece-Oran	AttributeSelecedetClassifier	Sınıf	% 14.44
Kelime-Formül	AdaBoostM1(SMO)	Sınıf	% 14.44
Kelime-Formül	LogitBoost	Sınıf	% 14.44
Kelime-Formül	SMO	Sınıf	% 14.44
Kelime-Formül	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.44
Kelime-Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	% 14.44
Kelime-Oran	Bagging(SMO)	Sınıf	% 14.44
Oran	AttributeSelecedetClassifier	Sınıf	% 14.44
Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.44
Ateşman	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.07
Formül-Oran	FilteredClassifier	Sınıf	% 14.07
Formül-Oran	J48	Sınıf	% 14.07
Formül-Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.07
Hece-Formül	AdaBoostM1(HoeffdingTree)	Sınıf	% 14.07
Hece-Formül	HoeffdingTree	Sınıf	% 14.07
Hece-Formül-Oran	AdaBoostM1(HoeffdingTree)	Sınıf	% 14.07
Hece-Formül-Oran	Bagging(HoeffdingTree)	Sınıf	% 14.07
Hece-Formül-Oran	FilteredClassifier	Sınıf	% 14.07
Hece-Formül-Oran	HoeffdingTree	Sınıf	% 14.07
Hece-Formül-Oran	SimpleLogistic	Sınıf	% 14.07
Hece-Kelime	RandomForest	Sınıf	% 14.07
Hece-Kelime-Formül	Bagging(Default)	Sınıf	% 14.07
Hece-Kelime-Formül-Oran	ClassificationViaRegression	Sınıf	% 14.07
Hece-Oran	AdaBoostM1(NaiveBayes)	Sınıf	% 14.07
Hece-Oran	FilteredClassifier	Sınıf	% 14.07
Hece-Oran	NaiveBayes	Sınıf	% 14.07
Kelime	Stacking(NB_HT_SVM_SMO)	Sınıf	% 14.07
Kelime-Oran	MultilayerPerceptron	Sınıf	% 14.07
Oran	FilteredClassifier	Sınıf	% 14.07
Formül	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	% 13.70
Formül-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Formül-Oran	ClassificationViaRegression	Sınıf	% 13.70
Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	% 13.70
Hece-Formül	RandomForest	Sınıf	% 13.70
Hece-Formül	Stacking(NB_HT_SVM_SMO)	Sınıf	% 13.70
Hece-Formül-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Hece-Formül-Oran	Bagging(SMO)	Sınıf	% 13.70
Hece-Formül-Oran	LogitBoost	Sınıf	% 13.70
Hece-Kelime	AdaBoostM1(SMO)	Sınıf	% 13.70
Hece-Kelime	SMO	Sınıf	% 13.70
Hece-Kelime-Formül	Bagging(NaiveBayes)	Sınıf	% 13.70
Hece-Kelime-Formül	J48	Sınıf	% 13.70
Hece-Kelime-Formül-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Hece-Kelime-Formül-Oran	FilteredClassifier	Sınıf	% 13.70
Hece-Kelime-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Hece-Kelime-Oran	FilteredClassifier	Sınıf	% 13.70
Hece-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Hece-Oran	Bagging(NaiveBayes)	Sınıf	% 13.70
Kelime	RandomForest	Sınıf	% 13.70
Kelime-Formül-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Kelime-Formül-Oran	FilteredClassifier	Sınıf	% 13.70
Kelime-Formül-Oran	MultilayerPerceptron	Sınıf	% 13.70
Kelime-Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Kelime-Oran	FilteredClassifier	Sınıf	% 13.70
Oran	AdaBoostM1(Default)	Sınıf	% 13.70
Ari	ClassificationViaRegression	Sınıf	% 13.33
Ateşman	AdaBoostM1(Default)	Sınıf	% 13.33

Çetinkaya-Uzun	Bagging(HoeffdingTree)	Sınıf	% 13.33
Çetinkaya-Uzun	MultilayerPerceptron	Sınıf	% 13.33
Formül	AdaBoostM1(Default)	Sınıf	% 13.33
Hece-Formül	AdaBoostM1(Default)	Sınıf	% 13.33
Hece-Formül	Bagging(HoeffdingTree)	Sınıf	% 13.33
Hece-Kelime	AdaBoostM1(Default)	Sınıf	% 13.33
Hece-Kelime	FilteredClassifier	Sınıf	% 13.33
Hece-Kelime-Formül	AdaBoostM1(Default)	Sınıf	% 13.33
Hece-Kelime-Formül	FilteredClassifier	Sınıf	% 13.33
Kelime	AdaBoostM1(Default)	Sınıf	% 13.33
Kelime	Bagging(HoeffdingTree)	Sınıf	% 13.33
Kelime	FilteredClassifier	Sınıf	% 13.33
Kelime-Formül	AdaBoostM1(Default)	Sınıf	% 13.33
Kelime-Formül	FilteredClassifier	Sınıf	% 13.33
Kelime-Formül-Oran	Bagging(SMO)	Sınıf	% 13.33
Kelime-Oran	J48	Sınıf	% 13.33
Kelime-Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	% 13.33
Oran	MultilayerPerceptron	Sınıf	% 13.33
Çetinkaya-Uzun	AdaBoostM1(NaiveBayes)	Sınıf	% 12.96
Çetinkaya-Uzun	NaiveBayes	Sınıf	% 12.96
Çetinkaya-Uzun	SimpleLogistic	Sınıf	% 12.96
Formül-Oran	MultilayerPerceptron	Sınıf	% 12.96
Hece-Formül	Bagging(SMO)	Sınıf	% 12.96
Hece-Formül-Oran	Bagging(NaiveBayes)	Sınıf	% 12.96
Hece-Formül-Oran	MultilayerPerceptron	Sınıf	% 12.96
Hece-Kelime-Formül	AdaBoostM1(NaiveBayes)	Sınıf	% 12.96
Hece-Kelime-Formül	MultilayerPerceptron	Sınıf	% 12.96
Hece-Kelime-Formül	NaiveBayes	Sınıf	% 12.96
Hece-Oran	ClassificationViaRegression	Sınıf	% 12.96
Kelime	Bagging(NaiveBayes)	Sınıf	% 12.96
Kelime-Formül	J48	Sınıf	% 12.96
Kelime-Oran	LogitBoost	Sınıf	% 12.96
Oran	SimpleLogistic	Sınıf	% 12.96
Ari	Bagging(HoeffdingTree)	Sınıf	% 12.59
Ateşman	AttributeSelectedClassifier	Sınıf	% 12.59
Ateşman	J48	Sınıf	% 12.59
Çetinkaya-Uzun	AdaBoostM1(Default)	Sınıf	% 12.59
Formül	AttributeSelectedClassifier	Sınıf	% 12.59
Formül	RandomForest	Sınıf	% 12.59
Formül	SimpleLogistic	Sınıf	% 12.59
Hece-Formül-Oran	AdaBoostM1(NaiveBayes)	Sınıf	% 12.59
Hece-Formül-Oran	NaiveBayes	Sınıf	% 12.59
Hece-Kelime	J48	Sınıf	% 12.59
Hece-Oran	Bagging(SMO)	Sınıf	% 12.59
Hece-Oran	J48	Sınıf	% 12.59
Kelime-Formül-Oran	LogitBoost	Sınıf	% 12.59
Kelime-Oran	AdaBoostM1(SMO)	Sınıf	% 12.59
Kelime-Oran	SMO	Sınıf	% 12.59
Oran	J48	Sınıf	% 12.59
Oran	LogitBoost	Sınıf	% 12.59
Ari	AdaBoostM1(Default)	Sınıf	% 12.22
Ari	SimpleLogistic	Sınıf	% 12.22
Hece-Formül-Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	% 12.22
Hece-Kelime	AdaBoostM1(NaiveBayes)	Sınıf	% 12.22
Hece-Kelime	Bagging(NaiveBayes)	Sınıf	% 12.22
Hece-Kelime	NaiveBayes	Sınıf	% 12.22
Hece-Kelime-Formül	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	% 12.22
Hece-Kelime-Formül-Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	% 12.22
Ateşman	Bagging(Default)	Sınıf	% 11.85

Ateşman	MultilayerPerceptron	Sınıf	%11.85
Çetinkaya-Uzun	Bagging(NaiveBayes)	Sınıf	%11.85
Çetinkaya-Uzun	ClassificationViaRegression	Sınıf	%11.85
Çetinkaya-Uzun	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%11.85
Formül	J48	Sınıf	%11.85
Hece-Formül-Oran	AdaBoostM1(SMO)	Sınıf	%11.85
Hece-Formül-Oran	ClassificationViaRegression	Sınıf	%11.85
Hece-Formül-Oran	J48	Sınıf	%11.85
Hece-Formül-Oran	SMO	Sınıf	%11.85
Hece-Kelime	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%11.85
Hece-Kelime-Oran	LogitBoost	Sınıf	%11.85
Kelime	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%11.85
Kelime-Formül-Oran	AdaBoostM1(SMO)	Sınıf	%11.85
Kelime-Formül-Oran	SMO	Sınıf	%11.85
Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%11.85
Ari	Bagging(NaiveBayes)	Sınıf	%11.48
Ateşman	RandomForest	Sınıf	%11.48
Çetinkaya-Uzun	RandomForest	Sınıf	%11.48
Formül	LogitBoost	Sınıf	%11.48
Formül	MultilayerPerceptron	Sınıf	%11.48
Formül-Oran	Bagging(SMO)	Sınıf	%11.48
Hece-Formül	AdaBoostM1(SMO)	Sınıf	%11.48
Hece-Formül	LogitBoost	Sınıf	%11.48
Hece-Formül	SMO	Sınıf	%11.48
Hece-Kelime	Bagging(Default)	Sınıf	%11.48
Hece-Kelime-Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%11.48
Hece-Kelime-Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	%11.48
Kelime	AdaBoostM1(NaiveBayes)	Sınıf	%11.48
Kelime	NaiveBayes	Sınıf	%11.48
Ari	AdaBoostM1(HoeffdingTree)	Sınıf	%11.11
Ari	AttributeSelecedetClassifier	Sınıf	%11.11
Ari	HoeffdingTree	Sınıf	%11.11
Ari	J48	Sınıf	%11.11
Ari	LogitBoost	Sınıf	%11.11
Ari	MultilayerPerceptron	Sınıf	%11.11
Ateşman	AdaBoostM1(NaiveBayes)	Sınıf	%11.11
Ateşman	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%11.11
Ateşman	NaiveBayes	Sınıf	%11.11
Çetinkaya-Uzun	AttributeSelecedetClassifier	Sınıf	%11.11
Çetinkaya-Uzun	Bagging(Default)	Sınıf	%11.11
Çetinkaya-Uzun	J48	Sınıf	%11.11
Formül-Oran	LogitBoost	Sınıf	%11.11
Hece-Formül	Bagging(NaiveBayes)	Sınıf	%11.11
Hece-Kelime	AttributeSelecedetClassifier	Sınıf	%11.11
Hece-Kelime-Formül	AttributeSelecedetClassifier	Sınıf	%11.11
Hece-Oran	MultilayerPerceptron	Sınıf	%11.11
Kelime	AttributeSelecedetClassifier	Sınıf	%11.11
Kelime	Bagging(SMO)	Sınıf	%11.11
Kelime-Formül	AttributeSelecedetClassifier	Sınıf	%11.11
Ari	Bagging(Default)	Sınıf	%10.74
Ari	RandomForest	Sınıf	%10.74
Ateşman	Bagging(NaiveBayes)	Sınıf	%10.74
Ateşman	SimpleLogistic	Sınıf	%10.74
Formül	Bagging(SMO)	Sınıf	%10.74
Formül	ClassificationViaRegression	Sınıf	%10.74
Hece	AttributeSelecedetClassifier	Sınıf	%10.74
Hece-Formül	AttributeSelecedetClassifier	Sınıf	%10.74
Hece-Formül-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%10.74
Hece-Kelime	Stacking(NB_HT_SVM_SMO)	Sınıf	%10.74

Hece-Oran	AdaBoostM1(SMO)	Sınıf	%10.74
Hece-Oran	SMO	Sınıf	%10.74
Kelime	AdaBoostM1(HoeffdingTree)	Sınıf	%10.74
Kelime	HoeffdingTree	Sınıf	%10.74
Oran	Bagging(SMO)	Sınıf	%10.74
Ari	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%10.37
Formül	Bagging(Default)	Sınıf	%10.37
Hece	SimpleLogistic	Sınıf	%10.37
Hece-Kelime-Formül	LogitBoost	Sınıf	%10.37
Hece-Kelime-Formül-Oran	LogitBoost	Sınıf	%10.37
Hece-Kelime-Oran	J48	Sınıf	%10.37
Hece-Oran	LogitBoost	Sınıf	%10.37
Ateşman	AdaBoostM1(HoeffdingTree)	Sınıf	%10.00
Ateşman	HoeffdingTree	Sınıf	%10.00
Hece	J48	Sınıf	%10.00
Ateşman	ClassificationViaRegression	Sınıf	%9.63
Çetinkaya-Uzun	AdaBoostM1(HoeffdingTree)	Sınıf	%9.63
Çetinkaya-Uzun	HoeffdingTree	Sınıf	%9.63
Çetinkaya-Uzun	LogitBoost	Sınıf	%9.63
Hece-Formül	AdaBoostM1(NaiveBayes)	Sınıf	%9.63
Hece-Formül	J48	Sınıf	%9.63
Hece-Formül	NaiveBayes	Sınıf	%9.63
Hece-Kelime-Formül-Oran	AttributeSelectedClassifier	Sınıf	%9.63
Hece-Kelime-Formül-Oran	J48	Sınıf	%9.63
Hece-Kelime-Oran	AttributeSelectedClassifier	Sınıf	%9.63
Hece-Oran	Stacking(NB_HT_SVM_SMO)	Sınıf	%9.63
Kelime-Formül-Oran	AttributeSelectedClassifier	Sınıf	%9.63
Kelime-Oran	AttributeSelectedClassifier	Sınıf	%9.63
Hece	Bagging(HoeffdingTree)	Sınıf	%9.26
Hece	Stacking(NB_HT_SVM_SMO)	Sınıf	%9.26
Hece-Formül	Bagging(Default)	Sınıf	%9.26
Hece-Oran	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%9.26
Kelime	J48	Sınıf	%9.26
Ari	AdaBoostM1(NaiveBayes)	Sınıf	%8.89
Ari	NaiveBayes	Sınıf	%8.89
Ateşman	Bagging(SMO)	Sınıf	%8.89
Ateşman	LogitBoost	Sınıf	%8.89
Formül-Oran	LibSVM	Sınıf	%8.89
Hece	MultilayerPerceptron	Sınıf	%8.89
Hece	RandomForest	Sınıf	%8.89
Kelime-Formül-Oran	LibSVM	Sınıf	%8.89
Kelime-Oran	LibSVM	Sınıf	%8.89
Oran	ClassificationViaRegression	Sınıf	%8.89
Oran	LibSVM	Sınıf	%8.89
Çetinkaya-Uzun	Bagging(SMO)	Sınıf	%8.52
Kelime	AdaBoostM1(SMO)	Sınıf	%8.52
Kelime	SMO	Sınıf	%8.52
Ari	AdaBoostM1(SMO)	Sınıf	%8.15
Ari	Bagging(SMO)	Sınıf	%8.15
Ari	FilteredClassifier	Sınıf	%8.15
Ari	LibSVM	Sınıf	%8.15
Ari	SMO	Sınıf	%8.15
Ateşman	AdaBoostM1(SMO)	Sınıf	%8.15
Ateşman	FilteredClassifier	Sınıf	%8.15
Ateşman	LibSVM	Sınıf	%8.15
Ateşman	SMO	Sınıf	%8.15
Çetinkaya-Uzun	AdaBoostM1(SMO)	Sınıf	%8.15
Çetinkaya-Uzun	FilteredClassifier	Sınıf	%8.15
Çetinkaya-Uzun	LibSVM	Sınıf	%8.15

Çetinkaya-Uzun	SMO	Sınıf	%8.15
Formül	FilteredClassifier	Sınıf	%8.15
Formül	LibSVM	Sınıf	%8.15
Formül-Oran	AdaBoostM1(SMO)	Sınıf	%8.15
Formül-Oran	SMO	Sınıf	%8.15
Hece	AdaBoostM1(HoeffdingTree)	Sınıf	%8.15
Hece	Bagging(Default)	Sınıf	%8.15
Hece	FilteredClassifier	Sınıf	%8.15
Hece	HoeffdingTree	Sınıf	%8.15
Hece-Formül	FilteredClassifier	Sınıf	%8.15
Hece-Formül	LibSVM	Sınıf	%8.15
Hece-Formül	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%8.15
Hece-Formül-Oran	LibSVM	Sınıf	%8.15
Hece-Kelime	LibSVM	Sınıf	%8.15
Hece-Kelime-Formül	LibSVM	Sınıf	%8.15
Hece-Kelime-Formül-Oran	LibSVM	Sınıf	%8.15
Hece-Kelime-Oran	LibSVM	Sınıf	%8.15
Hece-Oran	LibSVM	Sınıf	%8.15
Kelime	LibSVM	Sınıf	%8.15
Kelime-Formül	LibSVM	Sınıf	%8.15
Hece	AdaBoostM1(SMO)	Sınıf	%7.41
Hece	Bagging(SMO)	Sınıf	%7.41
Hece	ClassificationViaRegression	Sınıf	%7.41
Hece	LibSVM	Sınıf	%7.41
Hece	SMO	Sınıf	%7.41
Oran	AdaBoostM1(SMO)	Sınıf	%7.41
Oran	SMO	Sınıf	%7.41
Formül	AdaBoostM1(SMO)	Sınıf	%7.04
Formül	SMO	Sınıf	%7.04
Hece	Bagging(NaiveBayes)	Sınıf	%6.67
Hece	LogitBoost	Sınıf	%6.67
Hece	AdaBoostM1(Default)	Sınıf	%5.93
Hece	AdaBoostM1(NaiveBayes)	Sınıf	%5.56
Hece	MajorityVoting(NB_HT_SVM_SMO)	Sınıf	%5.56
Hece	NaiveBayes	Sınıf	%5.56