YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

MELEZ YAKLAŞIMLARLA TÜRKÇE DOKÜMANLARDA YAZAR TANIMA

Filiz TÜRKOĞLU

FBE Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programında

Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Banu DİRİ

İSTANBUL, 2006

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ iv

ŞEKİL LİSTESİ v

ÇİZELGE LİSTESİ vi

ÖNSÖZ vii

ÖZET viii

ABSTRACT ix

1. GİRİŞ 1

2. YAZAR TANIMA ALANINDA YAPILMIŞ ÖNCEKİ ÇALIŞMALAR 3

3. YAZARLIK ÖZELLİKLERİ 8

3.1 Özellik Vektörleri 8
3.1.1 İstatistiksel Özellikler 8
3.1.2 Kelime Zenginliğine Dayalı Özellikler 9
3.1.3 Dilbilgisi Özellikleri 10
3.1.4 İşlevsel Kelimeler 11
3.2 Özellik Birleştirme Yoluyla Oluşturulan Vektörler 11
3.3 Özellik Azaltma Yoluyla Oluşturulan Vektörler 12
3.4 Külliyat 13
3.4.1 Külliyat - I 14
3.4.2 Külliyat - II 14
3.4.3 Külliyat - III 14

4. SINIFLANDIRMA YÖNTEMLERİ 15

4.1 Naive Bayes 15
4.2 Destek Vektör Makinesi – DVM (Support Vector Machine – SVM) 17
4.2.1 Bire Karşı Hepsi (One-Against Rest) 19
4.2.2 Bire Karşı Bir (Pairwise) 19
4.2.3 Yönlendirilmiş Çevrimsiz Çizge (Directed Acyclic Graph) 19
4.3 K-En Yakın Komşuluk – K-EK (K-Nearest Neighbour- K-NN) 19
4.4 Çok Katmanlı Algılayıcı – ÇKA (Multiple Layer Perceptron – MLP) 20
4.5 Öz Düzenleyici Özellik Haritası – ÖDH (Self Organizing Map – SOM) 21
4.6 Rastgele Orman – RO (Random Forest - RF) 23
4.7 Korelasyon Tabanlı Özellik Seçme 24
4.8 Oylama (Vote) 24

5. DENEYSEL SONUÇLAR 26

5.1 Külliyata Göre Deneysel Sonuçlar 26
5.1.1 Külliyat-I için Deneysel Sonuçlar 26
5.1.2 Külliyat-II için Deneysel Sonuçlar 28
5.1.3 Külliyat-III için Deneysel Sonuçlar 28
5.2 Yöntemlere Göre Deneysel Sonuçlar 31
5.2.1 Rastgele Orman 31
5.2.2 Naive Bayes 32
5.2.3 Destek Vektör Makinesi 33
5.2.4 Çok Katmanlı Algılayıcı 34
5.2.5 K-Enyakın Komşuluk 35
5.2.6 Öz Düzenleyici Özellik Haritası 36
5.3 Sınıflandırıcı Birleştirmeye Göre Deneysel Sonuçlar 37

6. SONUÇ 41

KISALTMA LİSTESİ

AADTT	Automatic Author Detection for Turkish Text
ÇKA	Çok Katmanlı Algılayıcı
DVM	Destek Vektör Makinesi
K-EK	K-Enyakın Komşuluk
KTÖS	Korelasyon Tabanlı Özellik Seçme
NB	Naive Bayes
ÖDH	Öz Düzenleyici Özellik Haritası
RO	Rastgele Orman
TKV	Türkçe Kelime Veritabanı

ŞEKİL LİSTESİ

Sayfa

Şekil 4.1 DVM yöntemi ile doğrusal sınıflandırma 18
Şekil 5.1 Rastgele Orman yöntemi ile her üç külliyattan elde edilen sınıflandırma başarı oranları 31
Şekil 5.2 Naive Bayes sınıflandırıcısı kullanılarak her üç külliyattan elde edilen başarı oranları. 32
Şekil 5.3 Destek Vektör Makinesi metodu ile her üç külliyattan elde edilen başarı oranları 33
Şekil 5.4 Çok katmanlı algılayıcı yöntemi ile deney yapıldığında külliyatlardan elde edilen başarı oranları 35
Şekil 5.5 K-Enyakın komşuluk ile sınıflandırma yapılarak her üç külliyattan elde edilen başarı oranları 36
Şekil 5.6 Öz düzenleyici özellik haritası yöntemi ile her üç külliyattan elde edilen başarı oranları 37

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Yazar tanıma alanında yapılan çalışmaların karşılaştırması 6
Çizelge 3.1 Özellik sayıları indirgenmiş olan yeni özellik vektörleri 13
Çizelge 3.2 Külliyat içerisinde yer alan yazarlar 13
Çizelge 5.1 Külliyat-I üzerinde elde edilen sınıflandırma başarıları 27
Çizelge 5.2 Külliyat-II üzerinde elde edilen sınıflandırma başarıları 29
Çizelge 5.3 Külliyat-III üzerinde elde edilen sınıflandırma başarıları 30
Çizelge 5.4 Külliyat-I için sınıflandırıcı birleştirme sonuçları 38
Çizelge 5.5 Külliyat-II için sınıflandırıcı birleştirme sonuçları 39
Çizelge 5.6 Külliyat-III için sınıflandırıcı birleştirme sonuçları 40
Çizelge 6.1 Yöntemlerden alınan en başarılı sonuçlar ve bu sonuçları veren özellik vektörleri 41
Çizelge 6.2 Sınıflandırıcı birleştirilerek elde edilen en iyi sonuçlar, külliyat ve özellik vektörleri 42

ÖNSÖZ

Yapay Zeka'nın bir dalı olan Doğal Dil İşleme alanında yapılmış olan bu çalışmanın amacı, yazarı belli olmayan dokümanların yazarını bulmaya çalışmaktadır. Çalışma, Türkçe dokümanlar üzerinde farklı özellikler çıkarılarak çeşitli sınıflandırma yöntemleriyle dokümanın ait olduğu sınıfı bulmaya dayanmaktadır. Ayrıca, bu özellik ve yöntemlerin birleştirilmiş halleri de sınıflandırma yapılırken analiz edilmiştir. Bu çalışma, Türkçe dokümanlar için yazar tanıma alanında yapılmış bu güne kadarki en kapsamlı çalışmadır.

Çalışmam boyunca hiç eksik etmediği motivasyonu ve zaman yönetimi, yönlendirmesi, esirgemediği yardımı için ve zamanını bana tüm titizliğiyle ayırdığı için Yrd.Doç.Dr.Banu Diri'ye teşekkür ederim.

Desteğini pratik çözümlere dönüştürerek yol almamdaki yardımları için Arş.Gör.M.Fatih Amasyalı'ya teşekkür ederim.

Bu uzun çalışma boyunca bana karşı hep anlayışlı ve sevgi dolu olan aileme ve arkadaşlarıma da sonsuz teşekkür ederim.

ÖZET

İnternet'in yaygınlaşmasıyla elektronik ortamdaki doküman sayısı oldukça artmıştır. Gittikçe artan bu bilgiye daha kolay ve hızlı erişmek amacıyla metin sınıflandırma önem kazanmaktadır. Son yıllarda, metin sınıflandırma alanında yapılan çalışmaların bir kısmı, yazar tanıma adı verilen ve anonim bir metnin yazarını veya yazarı şüpheli olan bir metnin yazarını belirlemeyi amaçlayan çalışmaları kapsamaktadır.

Bu çalışmada, Türkçe dokümanların yazarlarının belirlenmesinde farklı özelliklerin ve sınıflandırıcıların performansa etkileri araştırılmıştır. Dokümanların istatistiksel, dilbilgisel, kelime zenginliğine dayalı özellik vektörleri çıkarılmıştır. Ayrıca Türkçe dokümanlar için ilk defa, işlevsel kelimelerin frekansları çıkarılarak ayrı bir özellik vektörü daha oluşturulmuştur. Sonraki aşamada seçilen bazı vektörler birleştirilerek yeni özellik vektörleri oluşturulmuştur. Sistemin öğrenmesine etkisi olmayan veya ayırt edici özelliği fazla bulunmayan özellikleri elemek amacıyla, özelllik azaltma metodu uygulanarak yeni vektörler elde edilmiştir. Sonuçta, 14 farklı özellik vektörü oluşturulmuş ve bunlar ile denemeler yapılmıştır.

Kullanılan külliyat, sağlık, gündem, ekonomi gibi farklı konularda yazan 18 yazara ait, 35 adet doküman alınarak 630 metinden oluşmaktadır. Farklı doküman gruplarının, farklı konularda yazılan metinlerin ve yazar sayısının başarıya olan etkisini gözlemlemek amacıyla 3 farklı külliyat grubu oluşturulmuştur. Tüm deneylerde 10-kat çapraz geçerlilik uygulanmıştır.

Yazar belirlemede hangi özellik veya özellik birleşimlerinin daha başarılı olduğunu analiz etmek amacıyla altı farklı sınıflandırma metodu kullanılarak performansları karşılaştırılmıştır. Bu metodlar Naive Bayes, Destek Vektör Makinesi, Rastgele Orman, K-Enyakın Komşuluk, Çok Katmanlı Algılayıcı ve Öz Düzenleyici Özellik Haritası' dır. Sınıflandırıcı birleştirme işleminin performansını gözlemlemek amacıyla, Naive Bayes, Destek Vektör Makinesi ve Rastgele Orman yöntemleri birleştirilmiştir.

Yapılan denemelere göre, en başarılı sonuçlar, yazar sayısının az olduğu ve farklı konulardaki yazılardan oluşan külliyattan elde edilmiştir. Tüm özelliklerin birleştirilmesinden oluşan özellik vektörü, diğerlerine göre daha iyi performans göstermiş ve en yüksek başarı oranını Çok Katmanlı Algılayıcı yöntemi vermiştir. Birleştirilmiş sınıflandırıcılar ile bireysel sınıflandırıcılara göre daha düşük başarı sonuçları alınmıştır.

Anahtar kelimeler: Yazar tanıma, metin sınıflandırma, özellik seçme, sınıflandırıcı birleştirme, Naive Bayes, Destek Vektör Makinesi, Rastgele Orman, K-Enyakın Komşuluk, Çok Katmanlı Algılayıcı ve Öz Düzenleyici Özellik Haritası.

ABSTRACT

There are numerous text documents available in electronic form. With the rapid growth of online information, text categorization has become one of the best automated techniques for handling, organizing text data. During the last decades, many classification tasks that are called author attribution were studied for identifying the author of an anonymous text, or text whose authorship is in doubt.

In this study the effect of different features and classifiers on performance of author attribution of Turkish texts are explored. Different vectors of statistical, grammatical, richness features are generated. Also a set of function words were applied on Turkish documents for the first time. All feature sets are combined and new vectors are obtained. In order to escape from features that are not relevant and beneficial for learning, feature selection method is applied over features and new vectors are formed from these reduced features. In the end we obtained 14 different feature vectors.

Corpus used in this work is formed from singly-authored 630 documents obtained from 35 texts per 18 different authors that are writing on different subjects like medical, popular interest and economics. To determine the capability of identifying authorship for heterogeneous documents, and different dataset sizes, this corpus is divided into 3 parts: Dataset I, Dataset II, Dataset III. Experiments are run 10-fold cross-validation on all datasets.

To analyse which features or feature combinations are successful for identifying the author of a document, comparative performance of six different classification methods are used. These methods are Naive Bayes, Support Vector Machine, Random Forest, Multilayer Perceptron, k-Nearest Neighbour and Self-organizing Feature Vector. We combined Random Forest, Naive Bayes and Support Vector Machine in order to analyse success ratio in proportion to single classifiers.

According to experimental results, most successful results are obtained from corpus of which author count is less and documents are written on different topics. Feature vector which is combined from all features gives better performance than others. Highest score is obtained from Multilayer Perceptron method. Combined classifiers gave poor results in proportion to single classifiers.

Keywords: Authorship attribution, text classification, feature selection, combining classifier, Naive Bayes, Support Vector Machine, Random Forest, Multilayer Perceptron, K-Nearest Neighbour, Self-Organizing Feature Vector.

1. GİRİŞ

Web'in yaygınlaşması, bilgisayarların gelişmesi ile elektronik metinlerin sayısı her geçen gün artmaktadır. Bu elektronik verilere daha hızlı, kolay ve doğru bir şekilde erişebilmek için dokümanların otomatik olarak sınıflandırılması önem kazanmaktadır.

Yapay Zeka teknolojisi gün geçtikçe daha fazla hayatımıza girmekte ve daha ileri ürünlerle gelişmeye devam etmektedir. Bu teknolojilerden yapay sinir ağları, bilgisayarlara insanın özelliklerini kazandırmada ve bilgisayarın öğrenmesini sağlamaktadır. Yapay Zeka'nın bir kolu olan Doğal Dil İşleme "ana işlevi doğal bir dili çözümleme, yorumlama, anlama ve üretme olan bilgisayar sistemlerinin tasarımını ve gerçekleştirilmesini konu alan bilim ve mühendislik dalı" olarak tanımlanmaktadır.

Makine Öğrenmesi (Machine Learning) zaman içinde davranışların iyileştirilmesi olarak tanımlanmaktadır. Çok sayıdaki örneğin yorumlanması büyük boyutta insan emeği ve zaman gerektirir. Bu yorumlama maliyeti, Makine Öğrenmesi tekniklerini gerçek hayattaki Doğal Dil İşleme uygulamalarına uyarlamadaki büyük engellerden biridir.

Doküman sınıflandırma işlemi, dokümanların önceden tanımlanmış kategorilerle etiketlendirilmesidir. Doğal Dil İşlemede doküman sınıflandırmanın, e-posta filtreleme, yazar tanıma, web içeriği organizasyonu gibi birçok uygulama alanı vardır.

Bir dokümanı nitelendirmedeki iki ana faktör, dokümanın içeriği ve stilidir. Stilistik özellikler üzerine literatürde yapılan çalışmalar, dokümanın içeriği üzerine yapılan çalışmalarla kıyaslandığında oldukça sınırlı sayıda kalmaktadır. Bunun sebebi, stilin tam olarak tanımlanamaması ve Doğal Dil İşleme sistemlerine uyarlamanın zor olmasıdır. Stile dayalı işlemlerde istatistiksel metodların kullanımı gerçekçi bir yaklaşım olarak kanıtlanmıştır (Stamatatos, 2000).

Stil özelliklerine dayalı sınıflandırma iki şekildedir:

- Dokümanın türü (genre detection): Dokümanın türünü belirlemeyle ilgilidir
- Yazar tanıma (author attribution): Dokümanın yazarını belirlemeyle ilgilidir

Stil özelliklerine dayalı doküman sınıflandırma işleminin iki ana aşaması vardır:

- Stil özelliklerinin çıkarılması ve hesaplanması
- Sınıflandırma

Yazar tanıma, anonim veya yazarının kim olduğundan tam olarak emin olunamayan edebi dokümanların yazarlarının belirlenmesidir (Stamatatos, 2000). Aynı doküman üzerinde,

yazarlık iddia eden iki kişiden hangisinin dokümanın gerçek yazarı olduğunun tespitinde veya bir dokümanı yazdığını kabul etmeyen kişinin tespitinde, yazar tanıma uygulamalarından faydalanılır. Herhangi birinin çalışmasını gizlice alıp, kendi ismi altında yayınlamak zor değildir. Bu gibi durumlarda, yazar tanıma metodları dokümanın gerçek sahibini bulmakta önemli bir rol oynamaktadır. Yazar belirleme alanındaki çalışmalar bizim, dilleri ve insan zekasının nasıl çalıştığını anlamamızı sağlar (Gerritsen, 2003).

Öğrenme sistemlerinde değişik öğrenme stratejileri kullanılmaktadır. Bunlar eğiticili ve eğiticisiz olmak üzere ikiye ayrılır.

Eğiticili öğrenmede sistem, olayı bir eğitici yardımıyla öğrenir. Bu yöntemde sınıfların önceden bilinmesi ve bu sınıflara ait belgelerden oluşan bir eğitim kümesi gerekmektedir. Eğitici sistemde hem örnek için girdi, hem de o girdi sonucunda oluşması beklenen çıktı verilir. Sistemin yapması gereken girdileri, eğiticinin belirlediği çıktılara haritalamaktır.

Eğiticisiz öğrenmede ise, sınıfların önceden bilinmesine ve herhangi bir aşamada insan yardımına ihtiyaç yoktur. Sisteme sadece girdi değerleri gönderilir ve sistemin kendi kendine öğrenmesi beklenir. Sistemin çalıştırılıp sonlandırılmasından sonra çıktıların yorumlanmasının kullanıcı tarafından yapılması gerekmektedir.

Bu çalışmada yazar tanıma için eğiticili ve eğiticisiz temel yöntemler kullanılmaktadır. Eğiticili yöntemlerden Destek Vektör Makineleri (Support Vector Machine), Naive Bayes, K-Enyakın Komşuluk (K-Nearest Neigbour – K-NN), Çok Katmanlı Algılayıcı (Multilayer Perceptron) ve Rastgele Orman (Random Forest); eğiticisiz yöntemlerden de Öz Düzenleyici Özellik Haritası (Self Organizing Feature Map - SOFM) seçilmiştir. Öncelikle yazara ait özellikler çıkartılarak farklı özellik vektörleri elde edilmiş, ayrıca bu özellik vektörleri birleştirilerek yeni vektörler elde edilerek sınıflandırma başarıları gözlemlenmiştir. Daha sonra elde edilen özellik vektörleri kullanılarak özellik azaltıcı fonksiyonlar yardımıyla özellik sayısı indirgenmiş vektörler oluşturulmuş ve tüm vektörlerin başarı performansları birbirleri ile karşılaştırılmıştır. Çalışmanın sonucunda seçilmiş bazı sınıflandırıcılar birleştirilerek melez bir sistem oluşturulmuş ve başarı performansları ölçülmüştür.

Tezin ikinci bölümde, yazar tanıma alanında daha önce yapılmış olan çalışmalara yer verilmiştir. Üçüncü bölümde metinlerden elde edilen özellik vektörlerinin çıkarılması anlatılmış, dördüncü bölümde de bu çalışmada kullanılmış olan sınıflandırma yöntemlerinden ve bunların birleştirilmesinden bahsedilmiştir. Beşinci bölümde ise deneysel sonuçlara, külliyata ve yorumlara yer verilmiştir.

2. YAZAR TANIMA ALANINDA YAPILMIŞ ÖNCEKİ ÇALIŞMALAR

Yazar tanıma alanında yapılan çalışmaların sayısı son yıllarda hızla artmaktadır.

Stamatatos ve arkadaşları (Stamatatos, 1999), sözdizimsel stil özelliklerinin çeşitli kombinasyonlarını kullanarak dokümanların yazarlarını belirleme üzerine bir çalışma yapmışlardır. Bu sözdizimsel stil özellikleri, yayının yapıldığı 1999 tarihinde yazar tanıma için en ayırdedici özellikler olarak ileri sürülmüştür. Yapılan çalışmada Yunanca metinler incelenip bunlardan tamlama ve kelime ayrıştırıcılar elde edilmiştir. Cümle sayısı, kelime sayısı, noktalama işaretlerinin sayısı, isim tamlaması sayısı, isim tamlamasındaki kelime sayısı gibi özellikler çıkarılarak kullanılmıştır. Sözdizim ayrıştırmanın yanı sıra dokümanın ayrıştırılmayan kısmının miktarı da göz önüne alınarak yazarın sözdizimsel karmaşıklığına bakılmıştır. Tüm bu özellikler kullanılarak 22 özelliğe sahip bir özellik vektörü oluşturulmuştur. Sınıflandırmada Doğrusal Çoklu Regresyon metodu kullanılmıştır. Bu metod, her yazar için, bir dokümanın o yazar tarafından yazılma olasılığını üreten bir model oluşturması nedeniyle Markov modeline benzemektedir. Külliyat, bir Yunan gazetesinden alınan 10 farklı yazarın makalelerinden oluşmuştur. Her yazarın 20 adet yazısı alınmış, bunların 10 tanesi eğitim seti, 10 tanesi de test seti olarak kullanılmıştır. Deneylerde % 69'luk doğru sınıflandırma başarısı elde edilmiştir. Bu çalışmada, yazar tanıma ve yazar doğrulama problemleri birbirinden ayrılarak incelenmiştir. Yazar doğrulamanın amacı, birisi tarafından yazarı olduğu iddia edilen bir metnin o kişiye ait olup olmadığının güvenilirlik seviyesinin bulunmasıdır. Yazar tanımanın amacı ise, bir dokümanın önceden belirlenmiş yazar sınıflarından hangisine ait olunduğunun belirlenmesidir.

Peng ve arkadaşları, (Peng, 2003) geçen en sık kullanılan *l* adet n-gram'ları çıkarıp, her yazarı n-gram'lardan oluşan bir vektörle modelleyerek yazar tanımayı incelemişlerdir. *l* ve *n*'nin çeşitli değerleri test edilmiştir. Yazarı belli olan her doküman için özellik vektörü oluşturulduktan sonra sınıflandırma işlemi "en yakın komşuluk" algoritmasıyla gerçekleştirilmiştir. 19. ve 20. yüzyıla ait sekiz kurgu bilim yazarının dokümanlarından oluşturulan bir külliyat üzerinde, farklı *l* ve *n* değerleri denenerek, en sık geçen 20 unigram (n=1) kullanılarak, % 100 doğruluk oranı elde edilmiştir. Aynı metodlar ile 20 Yunanlı yazarın dokümanları kullanılarak yapılan yazar tanımada % 85'lik başarı elde edilmiş iken, Çince bir külliyat üzerinde bu başarı % 89'a ulaşmıştır. Bu çalışmada, farklı diller üzerinde yazar tanıma yapılarak savunulan metodun dilden bağımsız olduğu vurgulanmıştır.

Khmelev ve Tweedie (Khmelev ve Tweedie, 2001), Markov zincirlerini kullanarak bir metod

geliştirip yazar tanımayı incelemişlerdir. Metod, belirlenmiş bir dokümanın belirli bir yazar tarafından yazılma olasılığını elde etmek amacıyla Markov modelini kullanır. Markov modelinde en yüksek olasılık değerini veren yazar, kazanan yazar olup, o metni onun yazdığı kabul edilir. Karakter bi-gram'lar (n=2) kullanılarak, her yazar için Markov model elde edilir. 27x27'lik bir matriste her 1-gram'ın varlığı sayılarak her doküman için modeller oluşturulmuştur. Bu matriste her satır, o satırın toplamına bölünerek bir olasılık elde edilir. Çalışmada dokümanlar ön işlemlerden geçirilerek, noktalama işaretleri, gereksiz boşluklar ve büyük harfle başlayan kelimeler alınmamıştır. Kullanılan külliyat Gutenberg projesinden 45 yazarın 387 dokümanından, Federalist yayınlarından ve birçok yazarı olan "Outside the Cave of the Shadows" yayınından oluşmuştur. Gutenberg projesindeki yazarların sınıflandırılmasında Markov modelleri, dokümanların % 74,4'ünü doğru sınıflandırmıştır. "Outside the Cave of the Shadows" külliyatı altı parça şeklinde eğitilmiştir. Kaynağı belli olan parçalarda % 100 başarı elde edilmiş, Federalist yayınlarının sınıflandırılmasında ise % 91' lik başarı alınmıştır.

Fung, (Fung, 2003) Federalist yayınlarının yazarlık özelliklendirilmesi için Destek Vektör Makinesi sınıflandırıcısını kullanılmıştır. Çalışmada Federalist yayınlar "as", "of" ve "on" kelimelerinin üç boyutlu uzayında bir düzlemle ayrılmışlardır. Bir takım fonksiyonel kelimeler kullanılarak Destek Vektör Makinesi uygulanmış ve yayınlar birbirinden ayrılmıştır.

Diri ve Amasyalı, (Diri ve Amasyalı, 2003) Türkçe dokümanların yazarlarının belirlenmesi için melez bir sınıflandırıcı kullanmışlardır. Sınıflandırıcılardan biri dokümanın içeriğini, diğeri ise belirlenen 22 farklı stil özelliğini kullanmaktadır. Dokümanı içeriğine bağlı sınıflandırmada Naive Bayes, stil özelliklerine göre sınıflandırma da ise kendi geliştirdikleri Automatic Author Detection for Turkish Text (AADTT) metodunu kullanmışlardır. Külliyat, www.hurriyet.com.tr'den indirilen 18 yazara ait politika, güncel ve sağlık konulu dokümanlardan oluşup, denemeler iki farklı veri seti üzerinde yapılmıştır. Grup 1, 18 farklı yazarın 20 yazısı alınarak 360 dokümandan oluşup, 20 yazının 15 tanesi eğitim, 5 tanesi de test için kullanılmıştır. Grup II'de 18 farklı yazarın 35'şer yazısı alınarak 630 doküman elde edilmiş, bunlardan 35 yazının 28 tanesi eğitim, 7 tanesi de test için kullanılmıştır. Elde ettikleri sonuçlara göre stile dayalı özelliklerin daha başarılı olduğunu vurgulamışlardır. Naive Bayes ve stile dayalı özellikleri kullanan yöntem birleştirildiğinde tanıma başarısı yükselmiştir. Grup I üzerinde Naive Bayes'in başarısı % 43 iken, AADTT' nin başarısı % 84' tür. Grup II için Naive Bayes'in başarısı % 53 iken, AADTT'nin başarısı ise % 70'tir. Her iki

yöntem birleştirildiğinde Grup I için % 89, Grup II için % 82'lik başarı oranı elde edilmektedir. Çalışmalarında stile dayalı özelliklerin, ortak kelimelerin dokümanlarda geçme sıklığıyla bir arada kullanıldığında, tanıma başarısını arttırdığını söylemektedirler.

Koppel ve Schler (Koppel ve Schler, 2003), külliyatlarını bir e-posta tartışma grubundaki yazışmalardan oluşturarak çeşitli özellik vektörlerini ve bunların birleşimini kullanarak yazar özelliklendirme çalışması yapmıştır. Yaptıkları çalışma, De Vel (De Vel, 2001) tarafından yapılan e-posta metinleri üzerindeki yazar tanıma çalışmasına paralellik göstermektedir. Ancak Koppel ve Schler bu çalışmadan farklı, bir e-posta dokümanındaki yapısal özellikleri (attachment, HTML tag) kullanmamışlar, diğer özelliklere ek olarak kullanım ve imladan kaynaklanan farklı sistematik hataları da incelemişlerdir. Külliyatları, 11 yazar tarafından yazılan 480 e-postadan oluşmaktadır. E-postaların herbiri yaklaşık 200 kelimeden oluşmaktadır. Bu dokümanları işleyerek üç farklı özellik sınıfına göre değerler bulmuşlardır. Birinci sınıf, 480 tane fonksiyonel kelimenin metinlerdeki kullanım sıklığını göstermektedir. İkinci sınıfta, ikili kelime gruplarının kullanım sıklığına bakılmış ve metinde en az 3 kere geçen gruplar birer özellik olarak kullanılmıştır. Üçüncü sınıfta ise metinde geçen hatalara bakılmıştır. Bu hatalar kelime tekrarı, harf eksikliği, harf tekrarı, fazladan harf yazılması, zaman kipinin yanlışlığı vb. dir. Tüm denemeler 10-kat çapraz sağlama (k-fold cross validation) kullanılarak Destek Vektör Makinesi ve Karar Ağaçları yöntemleriyle test edilmiştir. En yüksek başarı Karar Ağaçları için % 72, Destek Vektör Makinesi ile de % 60 olarak elde edilmiştir.

De Vel, (De Vel, 2001) yazar tanıma çalışmasını farklı konulardaki e-posta metinleri üzerinde yapmış ve Destek Vektör Makinesi kullanarak bu metinleri sınıflandırmıştır. E-posta dokümanları üzerinde yazar tanıma çalışmalarının yapılması, yayınlar üzerinde veya başka tür kaynaklardan oluşturulmuş Külliyatlar üzerinde yazar tanıma çalışması yapmaktan daha farklıdır. Öncelikle, e-posta metinlerinin boyutları kısadır ve bilinen dil tabanlı ölçüm yöntemleri bunlar için uygun değildir. İkinci olarak, bir e-posta dokümanını formüle etmedeki kompozisyon stili genelde, aynı yazar tarafından yazılan normal metin dokümanlarından farklıdır. Çalışmada, normal metin dokümanlarından çıkarılan yazar profilinin, e-posta dokümanlarından çıkarılan yazar profilinden farklı olabildiği üzerinde durulmuştur. Kullanılan külliyat 3 farklı haber grubundan ve ana dili İngilizce olan 4 erkek yazardan alınan, 1259 dokümandan oluşturulmuş ve konusu olmayan e-posta metinleri çıkarılmıştır. Dokümanların gövde kısmı, ön işlemlerden geçirilerek yanıtlama metinleri ve imzalar elde edilmiştir. 170 tane stil özelliği ve 21 tane yapısal özellik oluşturulmuştur. Bunlar kısa

kelimelerin sayısı (“all”, ”at”, ”his”, vs.), ortalama cümle uzunluğu, kelime zenginliği, ortalama kelime uzunluğu, işlevsel kelimelerin sayısı, fonksiyonel kelime frekans dağılımı, kelimelerdeki karakterlerin sayısı, noktalama işaretleri olarak alınmıştır. Bu çalışmaların sonucunda en yüksek % 100, en düşük %17,2 başarı elde edilmiştir. Deneylerde kullanılan yazar sayısı oldukça azdır, ancak birçok yazardan farklı konularda çok sayıda e-posta toplamanın kolay olmadığı belirtilmiştir.

Çizelge 2.1’de yazar tanıma alanında son yıllarda yapılmış olan çalışmaların karşılaştırılması verilmiştir. Bu tabloda kimler, hangi yöntemleri kullanarak kaç sınıf içerisinden dokümanın yazarını tanıdıklarını, hangi doğal dilde çalıştıklarını, eğitim ve test setlerindeki doküman sayıları gösterilmiştir.

Çizelge 2.1 Yazar tanıma alanında yapılan çalışmaların karşılaştırması

<table>
<tr><th rowspan="2">Referans</th><th rowspan="2">Yöntem(özellik)</th><th rowspan="2">Yazar Sayısı</th><th>Eğitim seti</th><th>Test seti</th><th rowspan="2">Külliyat dili</th><th rowspan="2">Başarı oranı</th></tr>
<tr><th colspan="2">yazar başına metin sayısı</th></tr>
<tr><td>Stamatatos vd., 1999</td><td>22 stil özelliği</td><td>10</td><td>10</td><td>10</td><td>Yunanca</td><td>69%</td></tr>
<tr><td rowspan="4">Stamatatos vd., 2000</td><td>22 stil özelliği
Disk. Analiz</td><td rowspan="4">10</td><td rowspan="4">10</td><td rowspan="4">10</td><td rowspan="2">Yunanca
Veri Seti A</td><td>72%</td></tr>
<tr><td>22 stil özelliği
Çoklu Regresyon</td><td>66%</td></tr>
<tr><td>22 stil özelliği
Disk. Analiz</td><td rowspan="2">Yunanca
Veri Seti B</td><td>70%</td></tr>
<tr><td>22 stil özelliği
Çoklu Regresyon</td><td>69%</td></tr>
<tr><td rowspan="2">Stamatatos vd.,, 2001</td><td>En sık geçen 50 kelime</td><td rowspan="2">10</td><td rowspan="2">20</td><td rowspan="2">10</td><td rowspan="2">Yunanca</td><td>74%</td></tr>
<tr><td>22 stil özelliği</td><td>81%</td></tr>
<tr><td>De Vel, 2001</td><td>Yapısal ve dilsel özellik</td><td>4</td><td colspan="2">-</td><td>İngilizce</td><td>% 100</td></tr>
<tr><td>Khmelev, Tweedie 2001</td><td>Markov Model
Karakter bigram</td><td>45</td><td colspan="2">-</td><td>Gutenberg Projesi, İngilizce</td><td>% 74,4</td></tr>
</table>

		2	-		Federalist yayınları İngilizce	% 91
		2	-		"Outside the Cave of Shadow" yayını, İngilizce	% 100
Koppel ve Schler, 2003	Sözcüksel özellik, kelime grupları, hatalar	11	10-kat Çapraz Geçerlilik		İngilizce	% 100
Diri ve Amasyalı, 2003	Naive Bayes, tüm kelimeler	18	15	5	Türkçe Grup 1	43%
	AADTT 11 stil özelliği					84%
	NB+AADTT					89%
	Naive Bayes, tüm kelimeler	18	28	7	Türkçe Grup 2	53%
	AADTT 22 stil özelliği					70%
	NB+AADTT					82%
Peng vd., 2003	6-gram model	8	-	-	İngilizce	100%
	5-gram model				Çince	89%
	3-gram model				Yunanca	% 97
Halteren, 2004	Dilsel özellik	8	2 roman	20 roman	Hollandaca	99,4%
			9-kat Çapraz Geçerlilik			

3. YAZARLIK ÖZELLİKLERİ

Her yazarın kendisine özgü bir yazım üslubu vardır. Yazarların kendilerine özgü olan bu tarzına yazarlık özelliği (authorship attribution) denir. Yazarlara ait dokümanların öğrenme teknikleri kullanılarak sınıflandırılması için, bu dokümanlardan belirli özelliklerin elde edilerek, özel bir şekilde gösterilmesi gerekmektedir. Dokümanların birer özellik vektörü şeklinde gösterimi en çok kullanılan metodlardan biridir (Salton, 1975). Her doküman, boyutu *d* olan bir vektör ile ifade edilir. *d* boyutu, dokümanın *d* adet farklı özelliğe sahip olduğunu gösterir. Bu çalışmada dokümanlar için dört farklı özellik vektörü oluşturulmuştur.

3.1 Özellik Vektörleri

Bu çalışmada daha önceki yazar tanıma çalışmalarında ayrı birer özellik olarak kullanılan özellikler bir araya getirilerek dört farklı özellik vektörü oluşturulmuştur. Bu dört farklı özellik vektörü istatistiksel, dilbilgisi, kelime zenginliği ve sözcüksel özelliklerden oluşmaktadır.

3.1.1 İstatistiksel Özellikler

İlk üsluba dayalı çalışmalar bir dokümandaki özellikleri sayma fikri şeklinde ortaya çıkmıştır. Bu fikir, kelime uzunluklarını ve cümle uzunluklarını sayma şeklinde hayata geçirilmiştir (Diederich, 2000). Daha sonra kelime sayısı, cümle sayısı, bir kelimedeki harf sayısı, noktalama işaretleri sayısı gibi özellikler de istatistiksel özellikler olarak yazar tanıma çalışmalarında oldukça sık yer almaya başlamıştır. Graham ve arkadaşları (Graham vd., 2004) karakter ve hece cinsinden ortalama kelime uzunluğunu ve her kelime uzunluğunun frekansını hesaplamıştır. Kelih ve arkadaşları (Kelih vd., 2005) yazar sınıflandırmada kelime uzunluğunun etkisini incelemişlerdir. De Vel ve arkadaşları (De Vel vd., 2001) e-posta metinleri üzerinde yaptıkları çalışmada ortalama cümle uzunluğu, boş satırların tüm satırlara oranı, ortalama kelime uzunluğu, noktalama işaretleri sayısı gibi istatistiksel özellikleri kullanmışlardır. Diri ve Amasyalı (Diri ve Amasyalı, 2003), kelime sayısı, cümle sayısı, ortalama cümle uzunluğu, bir kelimedeki karakter sayısı, noktalama işaretleri sayısı, kelime başına hece sayısı vb. gibi 22 tane ayırt edici özellik belirlemiş ve kullanmışlardır.

Bu çalışmada 10 farklı istatistiksel özellik kullanılmıştır. Bunlar:

- Dokümanın uzunluğu (dokümandaki toplam kelime sayısı)
- Ortalama cümle uzunluğu (bir cümledeki kelime sayısı)
- Cümle sayısı

- Tüm dokümandaki nokta sayısı
- Tüm dokümandaki üç nokta sayısı
- Tüm dokümandaki virgül sayısı
- Tüm dokümandaki noktalı virgül sayısı
- Tüm dokümandaki iki nokta sayısı
- Tüm dokümandaki ünlem sayısı
- Tüm dokümandaki soru işareti sayısı

Çalışmada bu özellikler kullanılarak oluşturulan vektörden ***Vist*** olarak bahsedilecektir.

3.1.2 Kelime Zenginliğine Dayalı Özellikler

Bir yazarın kelime dağarcığının zenginliğini bulmak amacıyla farklı istatistiksel özellikler kullanılmıştır. Bu özellikler bir yazarın yaratıcılığını göstermektedir. Yazar tanıma çalışmalarındaki makalelerin bir çoğu kelimelere ve onların kullanım sıklığına dayanır (Burrows 1992, Holmes 1994, Forsyth 1995). Bu durum, az sayıda dokümanın kelime frekanslarının istatiksel analizini yapmanın ön işlemesiyle gerçekleştirilir.

Bu çalışmada yazarın kelime zenginliğini ölçmek amacıyla:

1. Farklı kelime sayısının toplam kelime sayısına oranı (type/token ratio)
2. Dokümanda yalnızca bir kere geçen kelimelerin sayısı (hapax legomena)
3. Dokümanda sadece iki defa geçen kelimelerin sayısı (hapax dislegomena)

çıkarılmış ve bu özellikler kullanılarak oluşturulan vektöre ***Vzen*** denilecektir.

Birinci özellik *V/N* olarak gösterilmektedir. *V* dokümandaki farklı sözcük sayısı, *N* dokümandaki toplam sözcük sayısıdır. Tüm bu özellikler, kelime dağarcığı açısından yazarın daha önce kullanmadığı bir yazım şeklini üretebilme olasığını hesaplar. En fazla sıklıkla kullanılan kelimelerin dokümanlarda geçmesi muhtemel olarak beklendiğinden, nadiren kullanılan kelimelerin bir yazarı ayırdetmek için daha önemli bir bilgi olduğu düşünülür.

De Vel ve arkadaşları (Del Vel vd., 2001, 2002) e-posta metinlerinin yazarlarını bulmak için yaptıkları çalışmalarında; Graham ve arkadaşları (Graham vd., 2003) stilistik özeliklerden faydalanarak dokümanları sınıflandırma çalışmalarında; Stamatatos ve arkadaşları (Stamatatos vd., 2000) doküman sınıflandırmayı hem tür hem de dokümanın yazarı açısından incelediklerinde; Koppel ve Schler (Koppel ve Schler, 2004), dokümanın yazarını doğrulama için yaptıkları çalışmalarında bu üç özelliği de kullanmışlardır.

3.1.3 Dilbilgisi Özellikleri

Bir metindeki isim, sıfat, fiil, zarf, zamir, bağlaç, ünlem, ortaç sayıları da ayırt edici özelliklerdir. Diri ve Amasyalı'nın (Diri ve Amasyalı, 2003), Türkçe dokümanlar üzerinde inceledikleri yazar tanımada oluşturdukları 22 özelliğin 8 tanesi bu kelime tiplerinin sayısının cümle sayısına oranından oluşmaktadır.

Bu çalışmada dilbilgisi özellikleri ***Vdilt*** ve ***Vdilc*** özellik vektörleri altında incelenmiştir. Vdilt adı verilen bu vektör, dokümanda geçen toplam kelime türü sayılarını tutmaktadır. Yani ilgili dokümanda kaç tane sıfat, isim, vb. olduğuyla ilgilenir. Vdilc ise, cümle başına düşen kelime tipi oranını tutar. Her tip için, tipin dokümandaki toplam sayısının dokümandaki toplam cümle sayısına oranı olarak alınır.

Bu özellikler çıkarılırken Türk Dil Kurumu'nun [9] 35000 kelimelik sözlüğünden yararlanılarak oluşturulan Türkçe Kelime Veritabanı (TKV) kullanılmıştır. Türkçede bir kelimenin tipi isim, sıfat, fiil, zarf, zamir, bağlaç, ünlem, ortaç tiplerinden biridir. Bununla birlikte, Türkçede aynı kelimenin birden fazla tipi de olabilir, yani aynı kelime hem sıfat hem de isim olarak kullanılabilinir. Örneğin, "*Melike' ye yeşil elbise çok yakışmış.*" cümlesindeki "*yeşil*" sıfattır. "*Melike yeşili sever.* " cümlesinde ise *"yeşil"* isim olarak kullanılmıştır. TKV'nda bir kelime için en az bir, en fazla üç farklı kelime tipi tutulmuştur. TKV'nda kelimenin sadece bir tipi varsa o tip seçilmektedir. Birden fazla tipe sahip olan bir kelimenin tipini bulmak için aşağıdaki kurallar (Diri ve Amasyalı, 2003) sırayla uygulanmıştır:

1. Bir kelimenin olası tiplerinden biri sıfatsa ve kelime ek almamışsa ve bir sonraki kelime isim veya zamirse bu kelime sıfattır.
2. Bir kelimenin olası tiplerinden biri sıfat ise ve kelime ek almamışsa, kelimenin olası tiplerinden sıfat çıkarılır ve tip sayısı azaltılır. Eğer tip sayısı 1'e düşüyorsa kelimenin tipi olarak alınır.
3. Bir kelimenin olası tiplerinden biri sıfatsa ve kelime ek almamışsa 7. madde uygulanır.
4. Bir kelimenin olası tipleri arasında sıfat yoksa, fakat zarf varsa ve bir sonraki kelime fiilse veya bir sonraki kelime cümlenin sonunda ise bu kelime zarf olarak alınır.
5. Bir kelimenin olası tipleri arasında sıfat yoksa, fakat zarf varsa ve bir sonraki kelime isimse, kelimenin olası tiplerinden zarf çıkarılır ve tip sayısı azaltılır. Eğer tip sayısı 1'e düşmüşse kelimenin tipi olarak alınır.
6. Bir kelimenin olası tipleri arasında sıfat yoksa, fakat zarf varsa ve kelime cümlenin sonundaysa bu kelime fiil olarak alınır.

Yukarıdaki kurallarla tipi belirlenemeyen kelime için olası tipler arasında kelimenin en fazla

sıklıkla kullanılan tipi tercih edilir.

3.1.4 İşlevsel Kelimeler

İşlevsel kelimeler (function word), herhangi bir sözcüksel anlamı olmayan veya belirsiz anlamı olan, fakat bir cümledeki kelimeler arasında dilbilgisel ilişkiler kurmaya yarayan kelimelerdir. İşlevsel kelime olarak adlandırılmayan kelimeler içerik kelimeleri (content word) veya sözcüksel (lexical) kelimeler adını alır. Bunlar isim, fiil, sıfat ve çoğu zarflardır. Bazı zarflar işlevsel kelimedir, örneğin "neden". Sözlükler, içerik kelimelerini açıkça tanımlar, fakat işlevsel kelimelerin genel kullanımlarını tanımlamazlar. Dilbilgisinde ise işlevsel kelimelerin kullanımı detaylıca anlatılır. İşlevsel kelimeler edat, zamir, yardımcı fiil veya bağlaç olabilir. İşlevsel kelimeler ek almış veya almamış da olabilir.

Yazar tanıma çalışmalarında, işlevsel kelimelerle yapılan ilk çalışma Burrows tarafından yapılmıştır (Burrows, 1987). Baayen ve arkadaşları (Baayen, 2002) 42 işlevsel kelime ve 8 adet noktalama işaretini kullanarak yazarları özelliklendirmişler ve % 81,5 başarı elde etmişlerdir. Kullandıkları külliyat üç farklı konu üzerine yazılan 72 öğrenci makalesinden oluşmaktadır. Aynı külliyatı kullanarak Juola ve Baayen (Juola ve Baayen, 2003) %87 başarı elde ederken, kullandıkları özellikler arasında 164 işlevsel kelime yer almaktadır. Binongo (Binongo, 2003) ele aldığı bir kitabın yazarının kim olduğunu incelemek amacıyla, yazar özelliklendirmede en çok kullanılan 50 işlevsel kelimeyi kullanmıştır. Holmes ve arkadaşları (Holmes, 2001), iki yazarın dokümanını birbirinden ayırmak amacıyla 50 işlevsel kelime kullanmıştır. Zhao ve Zobel (Zhao ve Zobel, 2005), işlevsel kelimelerin yazar tanımada ne kadar başarılı olduğunu analiz etmiştir. Argamon ve Levitan (Argamon ve Levitan, 2005), yazar özelliklendirmede işlevsel kelimelerin kullanışlılığını ölçmüşlerdir.

Türkçe için daha önce işlevsel kelimelerin listesi çıkarılmadığından, bu çalışma için TKV kullanılarak bir işlevsel kelimeler listesi oluşturulmuştur. TKV'ndan yapılan bir sorgulamayla tipi edat, zamir, yardımcı fiil, bağlaç olan kelimeler çıkarılmış ve daha sonra tüm külliyat üzerinde bu kelimelerin kullanım sıklığı hesaplanmıştır. Külliyatta bir kez bile kullanılmayan kelimeler çıkarılmış ve geriye külliyatta yer alan 620 adet işlevsel kelime bırakılmıştır. Bunlarla adına ***Visl*** denilen özellik vektörü oluşturulmuştur.

3.2 Özellik Birleştirme Yoluyla Oluşturulan Vektörler

Özellik vektörlerini ayrı ayrı analiz etmenin yanısıra birlikte de değerlendirmek amacıyla özellik vektörlerinin birleştirilmesi yoluna gidilmiştir. Elde edilen yeni özellik vektörleri

aşağıdaki gibidir:

Vgt→Vist + Vzen + Vdilt + Visl

Vgc→ Vist + Vzen + Vdilc + Visl

Vgist → Vist + Vzen + Vdilt

Vgt, dört ana özellik vektörünün birleştirilmesiyle elde edilmiştir. Dilbilgisi özellik vektörlerinden Vdilt, yani doküman bazındaki tip sayısı kullanılmış ve özellik sayısı 641'e yükselmiştir. ***Vgc*** de, Vgt gibi dört farklı özellik vektörünün birleştirilmesinden meydana gelmiştir. Tek farkı Vdilt yerine cümle bazlı olan Vdilc kullanılmıştır ve özellik sayısı 641'dir. ***Vgist*** istatistiksel, dilbilgisel ve zenginliğe dayalı özelliklerin birleştirilmesinden oluşturulmuştur. Oluşturulan bu vektör 21 özelliğe sahiptir.

3.3 Özellik Azaltma Yoluyla Oluşturulan Vektörler

Dokümanları tanımlamak amacıyla çıkartılmış olan özelliklerin bazıları dokümanlar için ayırt edici veya gerekli olmayabilir. Bu durum hedef modelin öğrenme kalitesini de düşürebilir. Özelliklerin bir alt kümesiyle benzer başarı sonuçları alınırken, fazla sayıda özellik kullanmamız sistemi yavaşlatabilir. Yüksek boyutlu özellik setinin başarıya etkisini ölçmek, gereksiz özellikler varsa bunları elemek ve tanıma sistemini buna göre analiz etmek amacıyla özellik azaltma metodu uygulanmıştır. Bu bölümde bahsedilen özellik azaltıcı olarak WEKA [1] paketi içerisinde yer alan CfsSubsetEval fonksiyonu kullanılmıştır.

Bu çalışmada dilbilgisi özellikleri Vdilt ve Vdilc özellik vektörleri altında incelenmiştir. Vdilt, dokümanda geçen toplam kelime türü sayılarını tutmakta yani o dokümanda kaç tane sıfat, isim, vb. olduğuyla ilgilenmektedir. Vdilc ise cümle başına düşen kelime tipi oranını tutmaktadır. Her tip için, tipin dokümandaki toplam sayısının, dokümandaki toplam cümle sayısına oranıdır.

Dilbilgisi özelliklerinin metinde geçme sıklıklarını tutan Vdilt vektöründeki özellik sayısı, CfsSubsetEval fonksiyonu kullanılarak azaltılmıştır. Özellik sayısı azaldığında sınıflandırma işleminin süreside kısalmıştır. Oluşan yeni özellik vektörü ***Vdilta*** olarak adlandırılmıştır.

Bir diğer dilbilgisi özelliklerinden oluşan Vdilc vektörüne de CfsSubsetEval fonksiyonu uygulandığında ***Vdilca*** elde edilmiştir.

İşlevsel 620 kelimeden oluşan Visl özellik vektörü, CfsSubsetEval fonksiyonu kullanılarak

özellikleri azaltıldığında ***Visla*** adı verilen yeni bir özellik vektörü elde edilmiştir.

Tüm özelliklerin birleştirilmesiyle oluşturulan Vgt ve Vgc vektörlerine CfsSubsetEval fonksiyonu uygulandığında sırasıyla ***Vgta*** ve ***Vgca*** vektörleri elde edilmiştir.

İstatistiksel, dilbilgisel ve zenginliğe dayalı özellik vektörlerini birleştirerek oluşturduğumuz Vgist özellik vektöründen de, özellik azaltıcı CfsSubsetEval fonksiyonu kullanılarak, ***Vgista*** özellik vektörü elde edilmiştir.

Yeni özellik vektörlerinin, farklı doküman gruplarına göre elde edilen özellik sayıları Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1 Özellik sayıları indirgenmiş olan yeni özellik vektörleri

Özellik Vektörü	**Ana Özellik Vektörü-Özellik Sayısı**	**Külliyat-I Özellik Sayısı**	**Külliyat-II Özellik Sayısı**	**Külliyat-III Özellik Sayısı**
Vdilta	Vdilt – 8	5	3	7
Vdilca	Vdilc – 8	7	4	7
Visla	Visl - 620	24	27	43
Vgta	Vgt - 641	27	16	31
Vgca	Vgc- 641	24	17	40
Vgista	Vgist - 21	15	12	12

3.4 Külliyat

Bu çalışmada, önceki çalışmalarla kıyaslama yapabilmek ve güvenirliği arttırmak amacıyla Diri ve Amasyalı (Diri ve Amasyalı, 2003) tarafından oluşturulan ve kendi yazar tanıma çalışmalarında kullanılan genişletilmiş külliyat tercih edilmiştir. Bu külliyat için, günlük gazetelerimizden olan Hürriyet [2], Vatan [3] ve Sabah'tan [4] ekonomi, güncel, sağlık ve magazin gibi konularda yazılar indirilmiştir. Mevcut külliyat 18 farklı yazarın her birine ait 35 farklı yazısından oluşan 630 adet dokümandan meydana gelmektedir. Bu yazarlar cizelge 3.2'de verilmektedir.

Yapılan deneysel sonuçları, tek bir külliyata bağlı kalmadan yorumlayabilmek için, farklı doküman gruplarının farklı konularda yazılan metinlerin ve yazar sayısının başarıya olan etkisini gözlemlemek amacıyla 3 farklı külliyat grubu oluşturulmuştur.

Çizelge 3.2 Külliyat içerisinde yer alan yazarlar

Kod	Yazar	Konu	KByte	Kod	Yazar	Konu	KByte
1	Ayşe Arman	Popüler	203	10	Fatih Altaylı	Gündem	148
2	Bekir Coşkun	Gündem	65,7	11	Gündüz Tezmen	Sağlık	128
3	Cüneyt Ülsever	Gündem	100	12	Hadi Uluengin	Dünya	133
4	Doğan Hızlan	Popüler	100	13	Muharrem Sarıkaya	Gündem	111
5	Erkan Çelebi	Popüler	143	14	Oktay Ekşi	Gündem	97,1
6	Emin Çölaşan	Gündem	139	15	Pakize Suda	Popüler	119
7	Ercan Kumcu	Ekonomi	115	16	Serdar Turgut	Popüler	132
8	Ertuğrul Özkök	Gündem	130	17	Tufan Türenç	Gündem	110
9	Erdal Sağlam	Ekonomi	148	18	Yalçın Bayer	Gündem	235

3.4.1 Külliyat - I

Farklı konularda yazan 18 yazarın her birine ait 35 farklı yazısından, 630 dokümandan, oluşturulmuştur. Bu konular popüler hayat, güncel, ekonomi, dünya ve sağlık üzerinedir. Bu gruptaki yazarların kodu 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18'dir.

3.4.2 Külliyat - II

Sadece güncel konularda yazan 9 yazarın 35 adet yazı alınarak, 315 adet dokümandan oluşturulmuştur. Bu grupta bulunan yazarların kodu 2, 3, 6, 8, 10, 13, 14, 17, 18'dir.

3.4.3 Külliyat - III

Farklı konularda yazan 9 yazarın 35 adet yazısı alınarak, 315 adet dokümandan oluşturulmuştur. Bu gruptaki yazılar popüler hayat, güncel, dünya, sağlık ve ekonomi üzerinedir. Yazarların kodu ise 1, 5, 7, 8, 9, 10, 11, 12, 14' tür.

4. SINIFLANDIRMA YÖNTEMLERİ

Bu çalışmada, WEKA içerisinde yer alan sınıflandırma yöntemlerinden beşi seçilerek kullanılmıştır. Bunlar Naive Bayes, Destek Vektör Makinesi, Rastgele Orman, K-Enyakın Komşuluk ve Çok Katmanlı Algılayıcı olup tüm denemelerde öntanımlı (default) parametre değerleri kullanılmıştır. Ayrıca, Matlab'in kütüphanesindeki Öz Düzenleyici Özellik Haritası fonksiyonu 10'lu çapraz geçerlilik uygun hale getirilerek kullanılmıştır.

4.1 Naive Bayes

Naive Bayes sınıflandırıcısı, doküman sınıflandırmada başarısı kanıtlanmış, popüler bir makine öğrenmesi tekniğidir (Domingos ve Pazzani, 1997). Naive Bayes sınıflandırıcısı, çok iyi bilinen, pratik, olasılığa dayanan, birçok uygulamada kullanılan bir sınıflayıcıdır.

Chakrabarti ve arkadaşları bir külliyattaki metinlerin hiyerarşik olarak organize edilmesi için Naive Bayes'i kullanmışlardır (Chakrabarti vd., 1997). Frietag ve McCallum, dokümanlardan bilgi çıkarabilmek için, ağın her düğümündeki kelime dağılımını hesaplamak amacıyla Naive Bayes benzeri bir model kullanmıştır (Frietag ve McCallum, 1999). Dumais ve arkadaşları doküman sınıflama işlemini otomatize etmek amacıyla Naive Bayes ve diğer metin sınıflayıcıları kullanmışlardır (Dumais vd., 1998). Naive Bayes' in oldukça sık kullanılmasının bir başka nedeni kolay anlaşılabilir bir yöntem olmasıdır. Bazı alanlarda Naive Bayes, performans olarak yapay sinir ağları ve karar ağacı öğrenmelerine karşı tercih edilebilir. Peng ve arkadaşları, Bayes teoremini kullanarak dokümanlardan elde ettikleri karakter bazlı n-gram'larla yazar tanıma uygulaması gerçekleştirmişlerdir (Peng vd., 2002). Domingos ve Pazzani, Naive Bayes' in hangi durumlarda sınıflandırma için en uygun olduğunu ve hangi durumlarda olasılık tayinlerinin (assessment) yanlış olduğunu araştırmışlardır (Domingos ve Pazzani, 1996). Domingos ve Pazzani bu noktayı netleştirip, Naive Bayes'in sınıflandırma için en uygun olduğu basit durumları göstermişlerdir.

Naive Bayes hedefe ulaşmak için en uygun imkanları seçmeye yönelik bir model oluşturur. Yeni örnekler üzerindeki sınıflandırma, Bayes kuralı kullanılarak, örneğe en çok benzeyen sınıf seçilerek yapılır. Naive Bayes sınıflandırıcısı örneklerin tüm özelliklerinin, sınıfın şartlarını veren özellikler açısından birbirinden bağımsız olduğunu kabul eder. Buna "Naive Bayes Bağımsızlık Varsayımı" denir. Bu varsayım birçok gerçek problemde geçersiz olsa da, Naive Bayes çoğu zaman çok iyi sınıflandırma performansı gösterir ve bu sebeple diğer Bayes sınıflandırıcılardan hesaplama zamanı olarak çok daha hızlı çalışır (Friedman, 1997; Domingos ve Pazzani, 1997). Bağımsızlık varsayımı ve her özellik için, parametreler ayrı

olarak öğrenilir ve bu durum genelde özellik sayısı çok olduğunda öğrenmeyi büyük ölçüde kolaylaştırır (McCallum vd., 1998)(Kim vd., 2003).

Naive Bayes sınıflandırıcısı, her *x* örneğinin bir özellik vektörüyle ifade edildiği ve hedef fonksiyon *f(x)*'in *V* sınırlı değer kümesinden herhangi bir değer alabildiği öğrenme işlerine uygulanır. Hedef fonksiyon için bir eğitim seti hazırlanır. $<a_1, a_2, \ldots a_n>$ şeklindeki özellik vektörüyle ifade edilen, yeni bir örnek alınır ve eğiticiden bu örneğin hedef değerini yani ait olduğu sınıfı tahmin etmesi istenir.

Yeni gelen örneğin sınıfının belirlenmesi, Bayes yaklaşımına göre, örneği anlatan $<a_1,a_2,\ldots a_n>$ özellik değerlerine göre en uygun hedef değerinin, v_{MAP} (maximum a posteriori hypothesis), atanmasıdır.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2 \ldots a_n) \tag{4.1}$$

Bayes teoremi kullanılarak (4.1)'deki denklem şöyle yazılabilir:

$$\begin{aligned} v_{MAP} &= \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n \mid v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\ v_{MAP} &= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n \mid v_j) P(v_j) \end{aligned} \tag{4.2}$$

Eğitim verisine dayanarak (4.1)'deki iki terim hesaplanır. Her $P(v_j)$ değeri, eğitim verisinde geçen, her hedef değer v_j'nin frekansı sayılarak kolaylıkla hesaplanabilmektedir. Bu terimlerin sayısının olası örnekler çarpı olası hedef değerleri sayısına eşit olması probleme neden olabilir. Bu nedenle, gerçekçi hesaplar elde etmek için her örneğin, örnek uzayında defalarca görülmesi gerekmektedir.

Naive Bayes sınıflandırıcısı, özellik değerlerinin, hedef değerden bağımsız olması kabulüne dayanır. Bir başka deyişle, $(a_1,a_2 \ldots a_n)$ bağlamının olasılığı, herbir özelliğin olasılıklarının ürünüdür:

$$P(a_1, a_2 \ldots a_n \mid v_j) = \prod_i P(a_i \mid v_j) \tag{4.3}$$

Bu denklemi (4.2)'de yerine koyarsak Naive Bayes sınıflandırma yaklaşımını elde etmiş oluruz:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j) \tag{4.4}$$

v_{NB}, Naive Bayes sınıflandırıcısının çıktısını, yani sınıfı gösterir. Bir Naive Bayes sınıflandırıcısında, eğitim verisinden hesaplanması gereken farklı $P(a_i|v_j)$ terimlerinin sayısı, farklı özellik değerleri çarpı farklı hedef değerleri sayısı kadardır.

Kısaca, Naive Bayes öğrenme metodu, çeşitli $P(v_j)$ ve $P(a_i|v_j)$ terimlerinin eğitim verisi üzerindeki frekanslarına dayanarak hesaplandığı bir öğrenme aşamasıdır. Bu hesaplanma bütünü, öğrenme hipoteziyle ilişkilidir. Bu hipotez, daha sonra (4.4)'teki denklem uygulanarak her yeni örneğin sınıflandırılması için kullanılır.

Naive Bayes öğrenmesi ve diğer öğrenme metodları arasındaki ilginç bir fark, olası hipotezlerin alanı üzerinde açıkça arama yapılmamasıdır (bu durumda, olası hipotezin alanı, çeşitli $P(v_j)$ ve $P(a_i|v_j)$ terimlerine atanan olası değerlerin alanıdır). Bunun yerine, hipotez arama yapmadan, basitçe eğitim örnekleri arasındaki çeşitli veri kombinasyonlarının frekansları sayılarak oluşturulur (Mitchell, 1997).

Naive Bayes, olasılıklı bir sınıflandırıcıdır ve verilen bir dokümanı sınıflandırmak için kelimelerin ve sınıfların olasılıklarını kullanır (Náther, 2005). Bu çalışmada özellikler kelime frekansları değildir, ve özelliklerin devam eden bir dağılımları vardır. Bu nedenle, yapılan deneylerde Naive Bayes' in WEKA [1] uygulaması kullanılmıştır. Bu uygulama John ve Langley'in (John, Langley 1995) çalışmalarına dayanmaktadır.

John ve Langley, yaptıkları çalışmada, parametrik olmayan yoğunluk hesabı için, normallik varsayımından vazgeçerek bunun yerine istatistiksel metodlar kullanmışlardır. Bir Naive Bayes sınıflandırıcısı için, çeşitli doğal ve yapay alanlarda, yoğunluk hesabının iki metodunu karşılaştırarak, deneysel sonuçlar elde etmişlerdir. Bu metodlardan biri, normalliği varsaymak ve koşullara bağlı her durumu bir Gaussian ile modellemektir. Diğeri ise, parametrik olmayan kernel yoğunluk hesabını kullanmaktır. Birçok doğal ve yapay verilerle yaptıkları bu deneyler sonucunda, hatanın çok büyük oranda azaldığını gözlemlemişler ve Bayes modelleri ile öğrenmede kernel hesabının faydalı bir araç olduğunu yorumlamışlardır.

4.2 Destek Vektör Makinesi – DVM (Support Vector Machine – SVM)

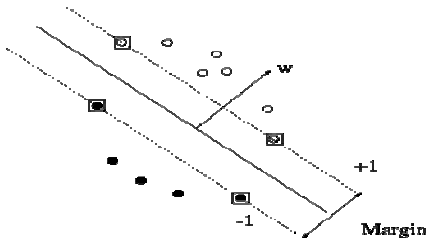
Destek Vektör Makinesi (DVM) yöntemi, Vapnik ve arkadaşları tarafından geliştirilmiş etkili bir öğrenme yöntemidir. DVM teorisi Vapnik'in çalışmasında geniş bir şekilde ele alınmıştır (Vapnik, 1995). Bu teori, yapısal risk minimizasyonu prensibine dayanır. Joachims (Joachims, 1998), dokümanları farklı konulara göre sınıflandırmak için DVM kullanmıştır. İstatistiksel özellik olarak dokümanda en az üç defa geçen kelimeleri kullanmışlar ve 10.000

özellik elde etmişlerdir.

Platt ve arkadaşları (Platt vd., 1998), doküman sınıflandırmada, doğru ve hızlı olduğu için doğrusal DVM kullandılar. Doğrusal DVM, test edilen sınıflandırıcılardan hızı kendisine en yakın olandan dahi 35 kat daha hızlıdır. Platt ve arkadaşları DVM sınıflandırıcısını, Reuter-21578 topluluğu, e-posta metinleri ve web sayfalarına uyguladılar.

Diederich ve arkadaşları (Diederich, 2000), yazar tanıma çalışmalarında DVM kullamışlardır. Alman gazetesinden alınan dokümanlarla çeşitli denemeler yapılmış ve % 60–80 oranında başarı elde etmişlerdir.

DVM, 2 sınıfa ait elemanların oluşturduğu eğitim kümelerini, en uygun düzlem ile birbirlerinden ayırmaya çalışan bir yöntemdir. En uygun düzlemi belirlerken mümkün olabildiğince az sayıda destek vektörü kullanılır. DVM ile doğrusal sınıflandırma yöntemi Şekil 4.1'de gösterildiği gibidir.



Şekil 4.1 DVM yöntemi ile doğrusal sınıflandırma

DVM yaklaşımı, yüksek boyutlu verilerin olduğu durumlarda dahi, güçlü bir genelleme özelliğine sahip, tahmine dayalı bir model geliştirebilir. DVM her zaman, bir optimizasyon çözümü aradığından fazla sayıda özelliğin analizine olanak tanıyabilmektedir [5].

Pozitif ve negatif örnekleri birbirinden ayıran bir düzlem bulunmaktadır ve bu düzlem üzerindeki noktalar $wx+b=0$ eşitliğini sağlamalıdır. Burada w, düzleme olan normal ve |b| / ||w|| düzlemden orijine dik uzaklıktır. Düzleme en yakın olan pozitif ve negatif örnekler arasındaki mesafe düzlemin *toleransı* olarak adlandırılırsa, DVM toleransın en büyük olduğu düzlemi bulmaya çalışmaktadır (Kepenekçi, 2004).

DVM yönteminin, diğer sınıflandırma yöntemlerinden en temel farkı, sadece iki sınıfı birbirinden ayırabilmesidir. Bu ayırım yapılırken dikkat edilmesi gereken en kritik nokta, iki sınıfa ait örnekleri birbirinden ayıracak olan düzlemin en uygun konumdan geçirilmesidir. Düzlemin uygun bir konumdan geçirilememesi durumunda, sınıflandırmada yanlış sonuçlar

üretilebilir.

DVM, yalnızca iki sınıfı karşılaştırmak üzere geliştirilmiş bir yöntemdir. Dolayısıyla, sistemde ikiden fazla sınıf olduğunda, DVM yöntemi ile çalışabilmek için gerekli düzenlemelerin yapılması gerekmektedir. Bu amaçla, ikiden fazla sınıfın olduğu bir sistemde, DVM yöntemini kullanabilmek için çeşitli algoritmalar geliştirilmiştir. Bu algoritmalar şunlardır:

4.2.1 Bire Karşı Hepsi (One-Against Rest)

Adından da anlaşıldığı gibi karşılaştırma sırasında sınıflardan biri (+) kabul edilirken, kalan diğer tüm sınıflar (-) olarak kabul edilir ve buna göre işlem yapılır. Burada, karşılaştırma sırasında + ve - sınıflardaki örnek sayıları farklı olabilmekte, aynı zamanda daha fazla karşılaştırma ve işlem gerektirdiğinden, fazla kullanılan bir yöntem değildir.

4.2.2 Bire Karşı Bir (Pairwise)

Sınıflar ikili gruplar halinde alınır ve bu ikililer üzerinden karşılaştırma yapılarak ilerlenir. Burada, ağaç üzerinde, en alttaki düğümden yukarıya doğru ilerlenir, en son adımda, kök düğüm ile karşılaştırma yapılıp sonuç bulunur. "Bire Karşı Hepsi" yöntemine kıyasla daha iyi bir çözüm olduğu söylenebilir.

4.2.3 Yönlendirilmiş Çevrimsiz Çizge (Directed Acyclic Graph)

Temel olarak, "Bire Karşı Bir" yöntemiyle benzerlik göstermektedir. Bu yöntemde, test verisi, ağaç üzerinde, kök düğümdeki ikiliden başlanarak karşılaştırılır ve ağacın en alt seviyesine kadar iteratif olarak ilerlenir. N seviyeli bir ağaç için önce (1,2) sınıf ikilisi kıyaslanır, bu ikiliden galip çıkacak sınıf *kazanan* ise; bir sonraki aşamada (kazanan,3) ikilisi kıyaslanır ve bu şekilde ilerlenerek, en sonunda (kazanan, N) ikilisi karşılaştırılıp, test verisinin hangi sınıfa daha yakın olduğu bulunur (Pal vd., 2004).

Bu çalışmada WEKA uygulaması içinde yer alan DVM sınıflandırıcısı kullanılmıştır. Bu sınıflandırıcı Platt (Platt, 1998) tarafından bahsedilen ardışık en küçük optimizasyon kullanılarak eğitilen Destek Vektör Makinesi'dir.

4.3 K-En Yakın Komşuluk – K-EK (K-Nearest Neighbour- K-NN)

K-Enyakın Komşuluk (K-EK), doküman sınıflandırmada kullanılan en popüler yöntemlerden biridir (Manning ve Schutze, 1999). Farklı külliyatlar üzerinde yapılan birçok araştırmada K-

Enyakın Komşuluk algoritmasının çok iyi performans gösterdiği gözlemlenmiştir (Yang ve Liu, 1999) (Joachism, 1998) (Baoli vd., 2002). Diederich'in (Diederich, 2000) yazar tanıma alanında yaptığı çalışmasında kullandığı sınıflandırma yöntemlerinden biri de K-Enyakın Komşuluk'tur. Zhao ve Zobel (Zhao ve Zobel, 2005), işlevsel kelimelerin yazarlık özelliklendirmede ne kadar başarılı olduğunu analiz etmek amacıyla çalışmalarında K-Enyakın Komşuluk metodunu da kullanmışlar ve %77 başarı elde etmişlerdir. Wolters ve Kirsten (Wolters ve Kirsten, 1999), dile bağlı özellikleri analiz etmek amacıyla K-Enyakın Komşuluk yöntemi ile sınıflandırma yapmışlardır.

K-Enyakın Komşuluk yönteminde en yakın k adet örnek bulunur ve bu örnekler en çok hangi sınıftan iseler, test örneği de o sınıfa aittir denir. Bu metodda, test örneğinin, eğitim kümesindeki bütün örneklere olan yakınlığı hesaplanır. Hesaplama işlemi için Öklid bağıntısı (4.5) kullanılır. Hesaplanan uzaklık değerleri, küçükten büyüğe doğru sıralanır. En yakın k adet örnek içinde en çok bulunan sınıf, test örneğinin ait olduğu sınıfı gösterir [6].

P ve Q noktaları için,

$P = (p_x)$ ve $Q = (q_x)$ ise Öklid bağıntısı şöyledir:

$$\sqrt{(p_x - q_x)}^2 = |p_x - q_x| \tag{4.5}$$

Bu çalışmada WEKA uygulaması içinde yer alan K-Enyakın Komşuluk sınıflandırıcısı kullanılmıştır. Bu sınıflandırıcı Aha ve Kibler (Aha ve Kibler, 1991) tarafından bahsedilen K-Enyakın Komşuluk yöntemi esas alınarak geliştirilmiştir.

4.4 Çok Katmanlı Algılayıcı – ÇKA (Multiple Layer Perceptron – MLP)

Bir yapay sinir ağının, öğrenmesi istenen olayların girdi ve çıktıları arasındaki ilişkiler doğrusal olmayan ilişkiler olduğunda, bu tür olaylar gelişmiş bir model olan Çok Katmanlı Algılayıcı (ÇKA) modeli ile öğrenilebilir. Rumelhart ve arkadaşlarının (Rumelhart vd., 1986) geliştirdiği bu modele *hata yayma modeli* veya *geriye yayım modeli (backpropogation network)* de denir. Özellikle sınıflandırma, tanıma ve genelleme yapmayı gerektriren problemler için çok önemli bir çözüm aracıdır.

ÇKA ağları ileriye doğru bağlantılıdır ve üç katmandan oluşur. Girdi katmanı, dış dünyadan gelen girdileri alarak ara katmana gönderir. Gelen her bilgi işlenmeden bir sonraki katmana gider. Ara katman, girdi katmanından gelen bilgileri işleyerek bir sonraki katmana gönderir. Bir ÇKA ağında birden fazla ara katman olabilir. Çıktı katmanı, ara katmandan gelen bilgileri

işleyerek, ağa girdi katmanından verilen girdilere karşılık ağın ürettiği çıktıları belirleyerek dış dünyaya gönderir.

ÇKA ağları, eğiticili öğrenme stratejisine göre çalışır. Ağlara eğitim sırasında hem girdiler hem de o girdilere karşılık üretilmesi beklenen çıktılar verilir. Ağın görevi her girdi için o girdiye karşılık gelen çıktıyı üretmektir. ÇKA ağlarının çalışması şöyledir:

- Ağın çözmesi istenen olay için daha önce gerçeklenmiş örnekler bulunur
- Ağın topolojik yapısı belirlenir. Kaç tane girdi ünitesi, kaç ara katman ve kaç çıktı elemanı olması gerektiği belirlenir
- Ağın öğrenme katsayısı, işleme giren elemanların toplama ve aktivasyon fonksiyonları, momentum katsayısı gibi parametreler belirlenir
- Elemanları birbirine bağlayan ağırlık değerlerine bağlangıç değerleri atanır
- Ağın öğrenmeye başlaması ve uygun olan ağırlıkları değiştirmesi için örnekler belirli bir düzeneğe göre ağa verilir
- Sunulan girdi için ağın çıktı değeri hesaplanır
- Ağın ürettiği hata değerleri hesaplanır
- Geri hesaplama yöntemi uygulanarak üretilen hatanın azaltılması için ağırlıkların değiştirilmesi gerçekleştirilir

Yukarıdaki adımlar ÇKA ağının öğrenmesi tamamlanıncaya kadar, yani gerçekleşen çıktı ile beklenen çıktılar arasındaki hatalar kabul edilir düzeye ininceye kadar devam eder. (Öztemel,2003)

De Vel (de Vel 1999), dokümanların sınıflandırma performansını ölçmek amacıyla Naive Bayes, K-Enyakın Komşuluk ve Destek Vektör Makinesi' nin yanı sıra ÇKA yöntemini de kullanmıştır. Diederich'in (Diederich, 2000) yazar tanıma alanında yaptığı çalışmasında ÇKA yöntemi ile de sınıflandırma yapılmış ve %93,3' lük bir başarı elde edilmiştir.

Bu çalışmadaki deneyler, WEKA'da ÇKA yönteminin öntanımlı parametreleri kullanılarak yapılmıştır.

4.5 Öz Düzenleyici Özellik Haritası – ÖDH (Self Organizing Map – SOM)

Öz Düzenleyici Özellik Haritası (ÖDH) Kohonen tarafından geliştirilmiştir. Kohonen ve arkadaşları (Kohonen vd., 2000), çok büyük sayıda doküman topluluğunu metinsel

benzerliklerini göz önüne alarak organize etmek amacıyla bir sistem geliştirmişlerdir. Bu sistem ÖDH yöntemi esas alınarak oluşturulmuştur. Özellik vektörü olarak dokümanlarda geçen kelimelerin istatistiksel gösterimleri kullanılmıştır. Çalışmalarının temel amacı, ÖDH yöntemini yüksek boyutlu verilerle ölçümlemektir. Deneylerinde 6.840.568 tane dokümanı, 1.002.240 düğümlü bir ÖDH sistemine yerleştirmişlerdir. Özellik vektörü olarak ağırlıklandırılmış kelime histogramından oluşturulan 500 boyutlu vektör kullanılmıştır.

ÖDH ağlarının girdi vektörlerini sınıflandırma ve girdi vektörlerinin dağılımını öğrenebilme yetenekleri çok yüksektir. Ağların en temel özelliği, olayları öğrenmek için bir öğretmen veya ağın üretmesi gereken çıktıların ağa bildirilme zorunluluğunun olmamasıdır. Özellikle beklenen çıktıların belirlenemediği problemler için kullanılır.

Ağ, girdi ve çıktı katmanından oluşmaktadır. ÖDH ağları yarışmayı kazanan elemanların 1, diğerlerinin 0 değerini alması ilkesine dayanır. Bir girdi verildiğinde, çıktı katmanındaki elemanlar birbirleri ile yarışırlar ve yarışmayı kazanan eleman girdinin sınıfını gösterir. Kazanan her elemanın komşularını gösteren komşuluk alanı oluşturulur. Komşuluk alanı içindeki bütün elemanlar kazanan eleman ile birlikte düşünülür ve komşuların eğitim sırasında ağırlıkları değişir.

Ağın eğitilmesi esnasında, herhangi bir *t* zamanında örnek setinden bir örnek, ağa verilir. Girdi vektörü X ile ağırlık vektörü A normalize edilmiş olmalıdır. Çıktı elemanlarından kazanan eleman bulunur. Bunun için iki yoldan birisi kullanılmaktadır:

1. Her elemanın çıktısı (*Ç*), ağırlıklarla girdilerin çarpımının toplamı ile bulunur. Yani,

$$Ç_i = \sum_i A_{i_0} X_i \tag{4.6}$$

Bu çıktı değerlerinden en yüksek değere sahip olan işlem elemanı yarışmayı kazanmaktadır. Bu elemanın *k*. eleman olması durumunda,

$$Ç_k = 1 \tag{4.7}$$

$$Ç_i = 0 \qquad i = 1, 2... \text{ ve } i \neq k \tag{4.8}$$

2. Öklid mesafesi (*d*) kullanılarak girdi vektörüne en yakın ağırlık vektörüne sahip eleman kazanan elemandır. İki vektör arasındaki mesafe:

$$d_j = \left\| X - A_j \right\| \tag{4.9}$$

(4.9)'daki gibidir. Her çıktı elemanı için bu mesafeler hesaplanır ve en küçük mesafesi olan eleman kazanan eleman olarak kabul edilir.

Kazanan eleman belirlendikten sonra bu eleman ve komşularının ağırlıkları (4.10)'daki formüle göre değiştirilir:

$$A(t+1)=A(t)+\lambda g(i,k)(X(t)-A(t)) \tag{4.10}$$

Burada λ öğrenme katsayısıdır. Öğrenme anında zamanla küçültülür (Öztemel, 2003). *g(i,k)* ise komşuluk fonksiyonudur, *i* ve *k* elemanlarının komşuluklarını belirler. *i* = *k* durumunda *g(i,k)* = 1 olur. Bu fonksiyon zaman içerisinde azalan bir fonksiyondur. Genel olarak *g(i,k)*,

$$g(k)=\exp(-\left\| d_i - d_k \right\|^2 / 2\sigma^2) \tag{4.11}$$

(4.11)'deki gibi ifade edilir. Formülde di ve dk, *i.* ve *k.* elemanların pozisyonunu gösteren vektörlerdir. σ ise komşuluk alanının genişliğini göstermektedir. Bu parametre de zaman içerisinde azaltılır (Öztemel, 2003).

Bu çalışmadaki ÖDH deneyleri, MATLAB SOM kütüphanesindeki fonksiyon kullanılarak çalıştırılmıştır.

4.6 Rastgele Orman – RO (Random Forest - RF)

Rastgele Orman, birçok karar ağacından oluşan ve bireysel ağaçlar tarafından oylanarak kazanan sınıfı veren bir yöntemdir. Bu yöntem Leo Breiman ve Adele Cutler tarafından geliştirilmiştir. Rastgele Orman yönteminde birçok sınıflandırma ağacı oluşturulur. Girdi vektöründeki bir objeyi sınıflandırmak için, ağaç kümesinde (forest) yer alan her ağaca giriş vektörleri verilir. Her ağaç bir sınıflandırma sonucu üretir. RO, tüm ağaçlar arasında en çok oyu alan sınıfı seçer. Her ağaç, eğitim setindeki örneklerin rastlantısal bir şekilde, yenisiyle değiştirme (replacement) mantığıyla oluşturulur.

Mevcut bir ağaç eğitim seti yenileme mantığıyla oluşturulurken, verinin üçte biri ağacın dışında kalır. Kalan kısım, ağaçların ağaç kümesine eklenirken sınıflandırma hatasını hesaplamak amacıyla kullanılır.

Ağaçların her biri oluşturulduktan sonra, tüm veri ağaç üzerinde çalıştırılır ve her veri çifti için yakınlık hesabı yapılır. Eğer çiftin her biri aynı son düğüm üzerine yerleşirse yakınlık

değerleri 1 arttırılır. Çalıştırma bittikten sonra yakınlık değerleri ağaç sayısına bölünerek normalizasyon yapılır [7].

Breiman (Breiman, 2001), Rastgele Orman üzerine yaptığı çalışmada bu yöntemin ağaç tahminlerinin bir birleşimi olduğunu belirtmiştir. Her ağaç, ağaç kümesindeki her ağaca aynı şekilde dağıtılan ve birbirinden bağımsız olarak örneklendirilen rastlantısal bir vektörün değerlerine dayanır. Breiman'a göre, ağaç sınıflandırıcılarından oluşan bir ağaç kümesinin genelleme hatası, ağaçların her birinin kümedeki bireysel güçlerine ve aralarındaki korelasyona bağlıdır.

Bu çalışmada, WEKA'da Breiman'ın (Breiman, 2001) çalışması esas alınarak oluşturulan fonksiyonlar varsayılan parametreler ile kullanılmıştır.

4.7 Korelasyon Tabanlı Özellik Seçme

Yakın zamandaki çalışmalar, özellik alt kümesi seçmenin makine öğrenmesi yöntemlerinin performansına olumlu bir etkisi olduğunu göstermiştir. Bazı algoritmalar, çok fazla veri yüzünden yavaşlarken ya da performansları kötü etkilenirken bazılarının öğrenme işlemiyle ilişkileri olmayabilir. Bu nedenle, özellik alt küme seçimi, öğrenme yöntemlerinin performansını arttıran bir yöntemdir (Hall ve Smith, 1997).

Korelasyon Tabanlı Özellik Seçme (KTÖS), korelasyona dayanan bir özellik alt küme seçme metodudur. KTÖS, hesaplama yapar ve özellik vektörlerinin kendilerindense özellik alt kümelerini derecelendirir. Bu yöntemin temelinde, özellik alt kümelerinin değerini hesaplama vardır. Bu hesaplamada, tek başına sınıf etiketini tahmin eden özelliklerin sayısına bakılırken, korelasyonun derecesi de göz önüne alınır (Hall, 1998).

KTÖS yöntemi oldukça esnektir, çeşitli araştırma ve değerlendirme metodlarının birleştirilmelerine imkan sağlar.

Bu çalışmadaki özellik alt kümeleri, WEKA'nın [1] KTÖS uygulaması ile oluşturulmuştur.

4.8 Oylama (Vote)

Bazı durumlarda, sadece bir sınıflandırıcıdan alınan sonuç verimli olmayabilir, ya da tek bir sınıflandırıcıya bağlı kalmak istenmeyebilir. Farklı veri grupları üzerinde sınıflandırıcıların başarısı farklılık gösterebilir. Bir veri grubunda başarılı olan bir yöntem, ikinci bir veri grubunda aynı başarıyı göstermeyebilir. Bu durumda sınıflandırıcıların birleştirilmesi gerekir. Başarı oranını ve doğruluğunu arttırmak amacıyla çeşitli sınıflandırıcılar birleştirilebilir.

Oylama yöntemi de bunlardan biridir.

Oylama, sayısal tahminleri veya olasılık değerlerini kullanarak sınıflandırıcıları birleştirme yöntemidir. Her sınıf için olasılık değerleri veya sayısal tahminler toplanır. En yüksek toplama sahip olan sınıf kazanan sınıftır.

5. DENEYSEL SONUÇLAR

Deneysel sonuçlar alınırken farklı alanlarda yazan 18 yazarın, 35 farklı yazısından oluşan üç farklı külliyat kullanılmıştır. Her bir yazıya ait 14 adet değişik özellik vektörü oluşturulmuştur. Özellik vektörlerinin yazar belirmedeki başarılarını test ederken 10'lu çapraz geçerlilik (10-fold cross validation) kullanılmıştır.

Çapraz geçerlilik, eğitim seti ile öğrenilen modelin başka yeni setler üzerinde nasıl performans göstereceğini hesaplayan yaklaşımlardam biridir. *x*'li çapraz geçerlilikte, veri *x* adet eşit alt kümeye bölünür. Her defasında eğitim setinden farklı bir alt küme çıkarılarak sistem *x* defa eğitilir [8].

Weka paketi içerisinde yer alan sınıflandırma yöntemlerini kullanırken (Naive Bayes, DVM, ÇKA, K-Enyakın Komşuluk ve RO) varsayılan parametreler tercih edilmiştir. K-Enyakın Komşuluk yöntemi ile sınıflandırma yapılırken çeşitli "k" değerleri denenmiş, bunlar arasında en başarılı olan k = 7 değeri seçilmiştir.

5.1 Külliyata Göre Deneysel Sonuçlar

Bu bölümde üç farklı külliyat üzerinde yapılan deneysel sonuçlar sırasıyla verilmiştir.

5.1.1 Külliyat-I için Deneysel Sonuçlar

İlk önce, popüler hayat, güncel, ekonomi, dünya ve sağlık gibi farklı konularda yazan 18 yazara ait 35 doküman alınarak oluşturulan Külliyat-I üzerinde, 14 farklı özellik vektörü kullanılarak 6 farklı sınıflandırıcıyla deneyler yapılmıştır. Çizelge 5.1'de 18 farklı yazarın sınıflandırılmasından alınan başarı sonuçları verilmiştir.

Külliyat-I'deki en iyi performans %82,1 olup istatistiksel, kelime zenginliği, dilbilgisel ve işlevsel kelime frekans özelliklerinden oluşturulan Vgt özellik vektörü ile DVM sınıflandırma yönteminden alınmıştır. DVM yönteminin, bu veri seti üzerinde özellik vektörünün boyutu arttıkça daha iyi sonuçlar verdiği gözlemlenmiştir. En kötü sonuç ise Visl ile ÇKA sınıflandırıcısı kullanıldığında alınmıştır. Başarı oranı % 9,2 olmuştur.

Tüm sınıflandırıcıların 14 özellik vektöründe verdikleri başarıların ortalamaları alındığında Naive Bayes'in %55,5 ile en başarılı, ÖDH'nın % 28,5 ile bu külliyatta en başarısız yöntem olduğu gözlenmiştir (Çizelge 5.1).

Çizelge 5.1 Külliyat-I üzerinde elde edilen sınıflandırma başarıları

Özellik Vektörü	RO%	NB%	DVM%	ÇKA%	K-EK%	ÖDH%	Ort Başarı%
Vist	60,3	55,5	41,9	59,6	56,3	31,2	50,8
Vzen	14,9	19,2	14,1	18,7	17,1	15,2	16,5
Vdilt	46	45,4	27,1	45,7	45,4	33,44	40,5
Vdilc	43,8	40,7	36,5	46,3	41,1	22,1	38,4
Visl	30,4	62,8	76,2	9,2	19,2	27,4	37,5
Vgt	49,6	67,7	82,1	10,6	25,5	29,2	44,1
Vgc	48	66,5	80	8,5	23,6	33,2	43,3
Vgist	71,4	68	65,9	71,4	68,6	38,3	63,9
Ort Başarı %	45,6	53,2	53	33,8	37,1	28,8	41,9
Vdilta	40,8	40,4	18,2	45,8	45,8	29,8	36,8
Vdilca	42,5	40,1	35,1	44,9	42,8	20,1	37,6
Visla	40	53	48,6	45	42,7	26	42,5
Vgta	70	74,1	72,9	72,7	64,7	27	63,6
Vgca	69,5	75,4	70,3	72,2	66,6	32	64,3
Vgista	67,9	67,5	60,5	71,7	65,4	34,6	61,3
Ort Başarı %	55,1	58,4	50,9	58,7	54,7	28,2	51
Genel Ort Başarı %	49,7	55,5	52,1	44,4	44,6	28,5	45,8

Özellik vektörlerinin her bir sınıflandırıcı için verdikleri doğruluk oranları toplanarak ortalaması alındığında en başarılı vektör Vgta iken, performansı en düşük vektör Vzen olmuştur (Çizelge 5.1).

Genel olarak Külliyat -I'in başarısı ortalama olarak %45,8'dir.

5.1.2 Külliyat-II için Deneysel Sonuçlar

Sadece güncel konularda yazan 9 yazara ait 35 doküman alınarak oluşturulan Külliyat-II üzerinde, 6 farklı sınıflandırıcı kullanılarak 14 farklı özellik vektörü ile deneyler yapılmıştır. Çizelge 5.2'de bu deneyler sonucunda elde edilen başarı sonuçları verilmiştir.

Külliyat-II'de %85 olan en yüksek başarı oranı, istatistiksel, kelime zenginliği, doküman bazlı dilbilgisel ve işlevsel kelime frekans özelliklerinden oluşturulan Vgc özellik vektörünün, ÇKA yöntemiyle sınıflandırması sonucunda elde edilmiştir. ÇKA yöntemi, bu veri seti üzerinde Külliyat-I'e göre daha iyi sonuçlar vermiştir. En kötü sonuç ise Vzen ile DVM sınıflandırıcısı kullanıldığında alınmıştır. Başarı oranı % 14,9 olmuştur.

Tüm sınıflandırıcıların 14 özellik vektöründe verdikleri başarıların ortalamaları alındığında ÇKA yöntemi %64,6 ile en iyi performansı, ÖDH yöntemi ise % 43,1 ile en kötü performansı veren yöntem olmuştur (Çizelge 5.2).

Özellik vektörlerinin ortalaması alındığında en başarılı vektör Vgca iken, performansı en düşük vektör Vzen'dir (Çizelge 5.2).

Genel olarak Külliyat-II'nin başarısı ortalama olarak % 56,7'dir.

5.1.3 Külliyat-III için Deneysel Sonuçlar

Popüler hayat, güncel, dünya, sağlık veya ekonomi konularında yazan 9 yazarın 35 dokümanı alınarak oluşturulan Külliyat-III üzerinde, 14 farklı özellik vektörü kullanılarak 6 farklı sınıflandırıcıyla deneyler yapılmıştır. Çizelge 5.3'te 9 farklı yazarın sınıflandırılmasından alınan başarı sonuçları verilmiştir.

Külliyat-III'deki en iyi performans %89,2 olup, istatistiksel, kelime zenginliği, doküman bazlı dilbilgisel ve işlevsel kelime frekans özelliklerinden oluşturulan Vgc özellik vektörü ile ÇKA sınıflandırma yönteminden alınmıştır. En kötü sonuç ise Vzen ile DVM sınıflandırıcısı kullanıldığında %28,2 olarak alınmıştır.

Tüm sınıflandırıcıların 14 özellik vektöründe verdikleri başarıların ortalamaları alındığında

Çizelge 5.2 Külliyat-II üzerinde elde edilen sınıflandırma başarıları

Özellik Vektörü	RO%	NB%	DVM%	ÇKA%	K-EK%	ÖDH%	Ort Başarı%
Vist	68,9	72,7	66,3	72,4	68,9	50	66,5
Vzen	15,5	19,3	14,9	15,5	17,5	18,42	16,9
Vdilt	63,5	64,8	54,6	61,2	57,1	55	59,4
Vdilc	40,3	41	42,5	42	39,4	33,64	39,8
Visl	41	58,7	77,1	77,7	25,4	39,81	53,3
Vgt	51,1	65,4	82,2	83,5	33,7	48,87	60,8
Vgc	56,2	65,7	83,8	85	34,2	53,09	63,0
Vgist	76,5	76,8	77,7	76,5	69,8	53,7	71,8
Ort Başarı %	51,6	58,1	62,4	64,2	43,3	44,1	53,9
Vdilta	55,9	58,4	42,2	59,4	57,5	46,1	53,3
Vdilca	35,5	41,6	36,2	36,5	36,5	30,2	36,1
Visla	60,0	67,3	69,5	63,2	52,0	41,3	58,9
Vgta	75,5	80,0	78,4	75,9	68,0	42,0	70,0
Vgca	78,4	84,1	79,7	81,0	73,3	41,1	72,9
Vgista	74,3	78,7	72,4	74,3	74,9	50,1	70,8
Ort Başarı %	63,3	68,4	63,1	65,1	60,4	41,8	60,3
Genel Ort Başarı %	56,6	62,5	62,7	64,6	50,6	43,1	56,7

Çizelge 5.3 Külliyat-III üzerinde elde edilen sınıflandırma başarıları

Özellik Vektörü	RO%	NB%	DVM%	ÇKA%	K-EK%	ÖDH%	Ort Başarı%
Vist	75	72	52,7	68	67,3	44	63,2
Vzen	33	29,8	28,2	34,6	33	30,9	31,6
Vdilt	64,1	53,7	30,8	60	58,4	45	52,0
Vdilc	54,9	60,3	55,5	60,6	55	42	54,7
Visl	53,3	76,2	82,5	87,9	31,1	46,7	63,0
Vgt	64,1	79	87,6	88,9	41	42,8	67,2
Vgc	65	78,4	87	89,2	35,2	50,5	67,6
Vgist	81	76,2	71,4	79,4	72,7	51	72,0
Ort Başarı %	61,3	65,7	62,0	71,1	49,2	44,1	58,9
Vdilta	58,4	54,3	30,5	60,7	58,4	42,1	50,7
Vdilca	57,1	59,0	56,2	60,0	59,4	39,8	55,3
Visla	65,0	74,0	75,9	75,0	59,0	48,5	66,2
Vgta	80,3	85,0	85,0	84,0	76,2	38,9	74,9
Vgca	79,4	85,0	86,0	84,1	73,0	48,1	75,9
Vgista	80,3	75,5	66,3	75,5	72,4	48,4	69,7
Ort Başarı %	70,1	72,1	66,7	73,2	66,4	44,3	65,5
Genel Ort Başarı %	65,1	68,5	64,0	72,0	56,6	44,2	61,7

ÇKA yönteminin %72 ile en başarılı, ÖDH yönteminin %44,2 ile en başarısız olduğu gözlenmiştir (Çizelge 5.3).

Özellik vektörlerinin ortalaması alındığında en başarılı vektör Vgta iken, performansı en düşük vektör Vzen olarak tespit edilmiştir (Çizelge 5.3).

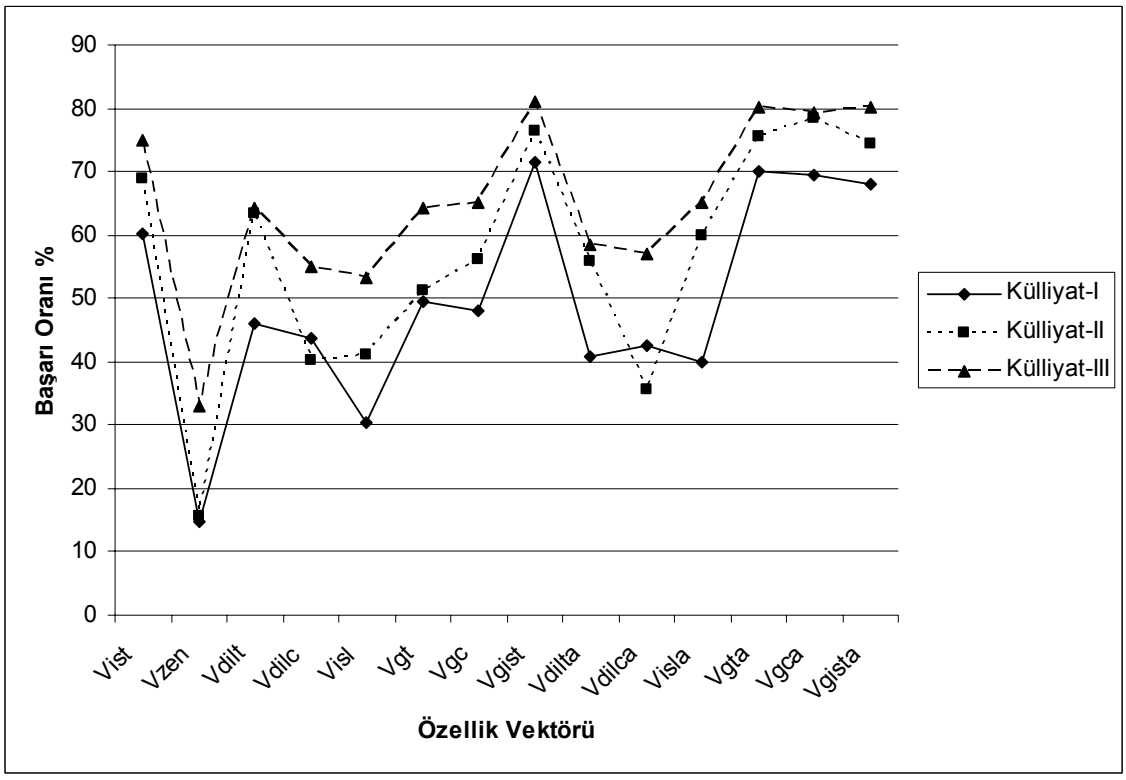
Genel olarak Külliyat-III'ün başarısı ortalama %61,7'dir.

5.2 Yöntemlere Göre Deneysel Sonuçlar

Bu bölümde kullandığımız altı yöntemin yazar tanımadaki başarıları ayrı ayrı verilmektedir.

5.2.1 Rastgele Orman

Rastgele Orman yöntemi ile her üç külliyattan elde edilen sınıflandırma başarı oranları Şekil 5.1'deki grafikte gösterilmiştir.



Şekil 5.1 Rastgele Orman yöntemi ile her üç külliyattan elde edilen sınıflandırma başarı oranları

Rastgele Orman yönteminin Külliyat-I üzerindeki en yüksek başarısı %71,4 olup Vgist özellik vektöründen elde edilmiştir. En kötü sonuç ise Vzen özellik vektörü ile alınan %14,9'dur.

Külliyat-II ile yapılan deneylerde bu yöntemin en iyi performansı Vgca ile alınan %78,4 sonucudur. En kötü performans ise Vzen özellik vektörü ile alınan %15,5'dir. Külliyat-III'te elde edilen en yüksek başarı %81 olup, Vgist özellik vektöründen elde edilmiştir. En kötü sonuç yine Vzen özellik vektöründen elde edilen % 33'tür.

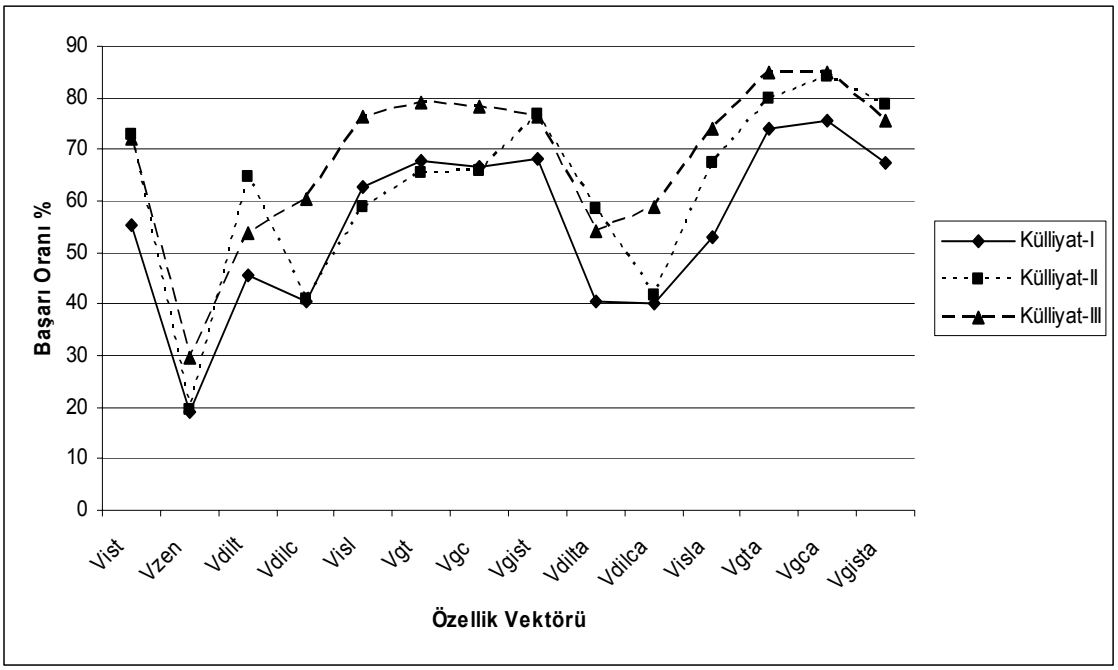
Özellik vektörlerine göre ortalama alındığında Rastgele Orman'ın en başarılı olduğu özellik vektörü %76,3 başarı ile Vgist'tür. Sınıflandırma için en elverişsiz özellik vektörü ise %21,1 ile Vzen' dir.

Her külliyat için başarı ortalamaları hesaplandığında, RO sınıflandırma yönteminin Külliyat-III'te, %65 ile en başarılı sonuç verdiği gözlemlenmiştir.

Genel olarak üç külliyat için Rastgele Orman yönteminin ortalama başarısı % 57,1'dir.

5.2.2 Naive Bayes

Naive Bayes sınıflandırıcısı kullanılarak her üç külliyattan elde edilen başarı sonuçları Şekil 5.2'deki grafikte gösterilmiştir.



Şekil 5.2 Naive Bayes sınıflandırıcısı kullanılarak her üç külliyattan elde edilen başarı oranları

Naive Bayes yönteminin Külliyat-I üzerindeki en yüksek başarısı %75,4 olup, Vgca özellik vektöründen elde edilmiştir. En kötü sonuç ise Vzen özellik vektörü ile alınan %19,2'dir. Külliyat-II ile yapılan deneylerde bu yöntemin en iyi performansı Vgca ile alınan %84,1'lik

sonuçtur. En kötü performans ise Vzen özellik vektörü ile alınan %19,3'tür. Külliyat-III'te alınan en yüksek başarı %85 olup, hem Vgta hem de Vgca özellik vektöründen elde edilmiştir. En kötü sonuç yine Vzen özellik vektöründen elde edilen %29,8'dir.

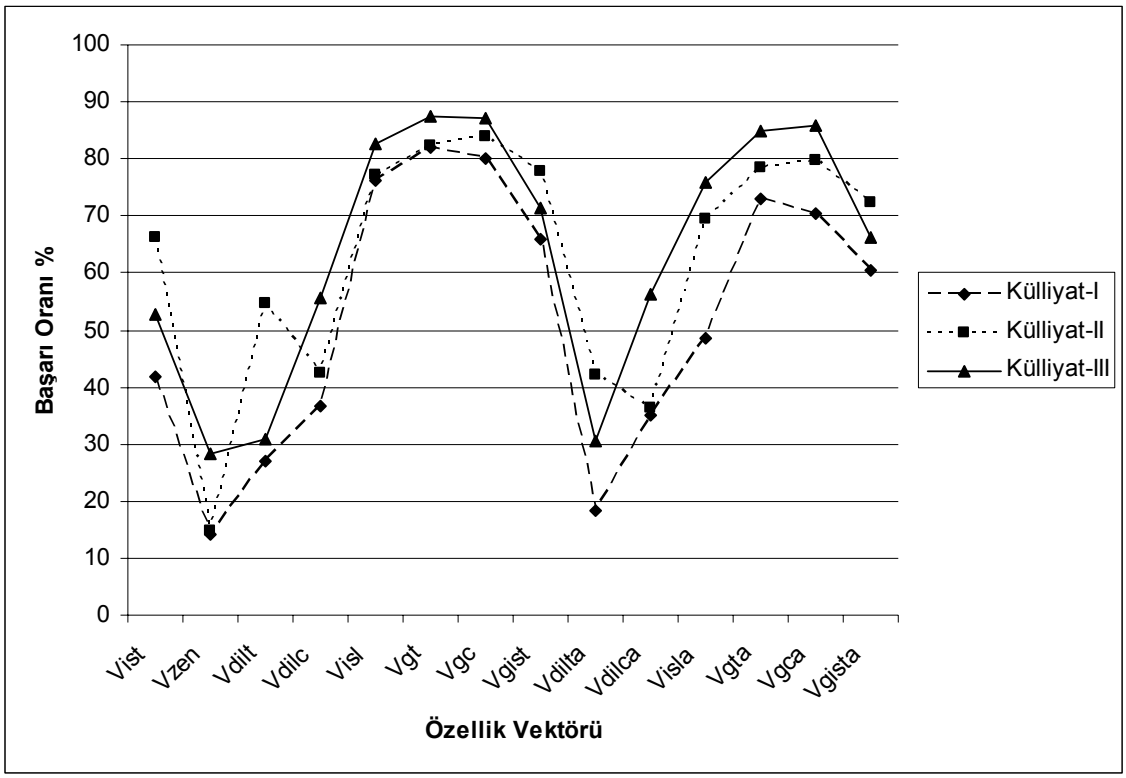
Özellik vektörlerine göre ortalama alındığında Naive Bayes'in en başarılı olduğu özellik vektörü %81 başarı ile Vgca'dır. Sınıflandırma için en elverişsiz özellik vektörü ise % 22,7 ile Vzen'dir.

Her külliyat için başarı ortalamaları hesaplandığında % 68,5'lik başarı oranıyla Külliyat-III, Naive Bayes'in en başarılı sonuç verdiği külliyattır.

Genel olarak üç külliyat için Naive Bayes yönteminin ortalama başarısı % 62,1'dir.

5.2.3 Destek Vektör Makinesi

Destek Vektör Makinesi metodu ile her üç külliyattan elde edilen başarı oranları Şekil 5.3'deki grafikte gösterilmiştir.



Şekil 5.3 Destek Vektör Makinesi metodu ile her üç külliyattan elde edilen başarı oranları

Destek Vektör Makinesi yöntemi ile Külliyat-I üzerinde yapılan sınıflandırmada en yüksek başarı %82,1 olup, Vgt özellik vektöründen elde edilmiştir. En kötü başarı sonucu ise Vzen

özellik vektörü ile alınan %14,1'dir. Külliyat-II ile yapılan deneylerde bu yöntemin en iyi performansı Vgc ile alınan %83,8 sonucudur. En kötü performans ise Vzen özellik vektörü ile alınan %14,9'dur. Külliyat-III'te alınan en yüksek başarı %87,6 olup, Vgt özellik vektöründen elde edilmiştir. En kötü sonuç yine Vzen özellik vektöründen elde edilen %28,2'dir.

Özellik vektörlerine göre ortalama alındığında DVM'nin en başarılı olduğu özellik vektörü %84 başarı ile Vgt'dir. Sınıflandırma için en elverişsiz özellik vektörü ise % 19'luk başarı ile Vzen'dir.

Her külliyat için başarı ortalamaları hesaplandığında Külliyat-III'ün, %64 başarı oranı ile Destek Vektör Makinesi'nin en başarılı olduğu külliyat olduğu gözlemlenmiştir.

Genel olarak üç külliyat için Destek Vektör Makinesi yönteminin ortalama başarısı % 59,6'dır.

5.2.4 Çok Katmanlı Algılayıcı

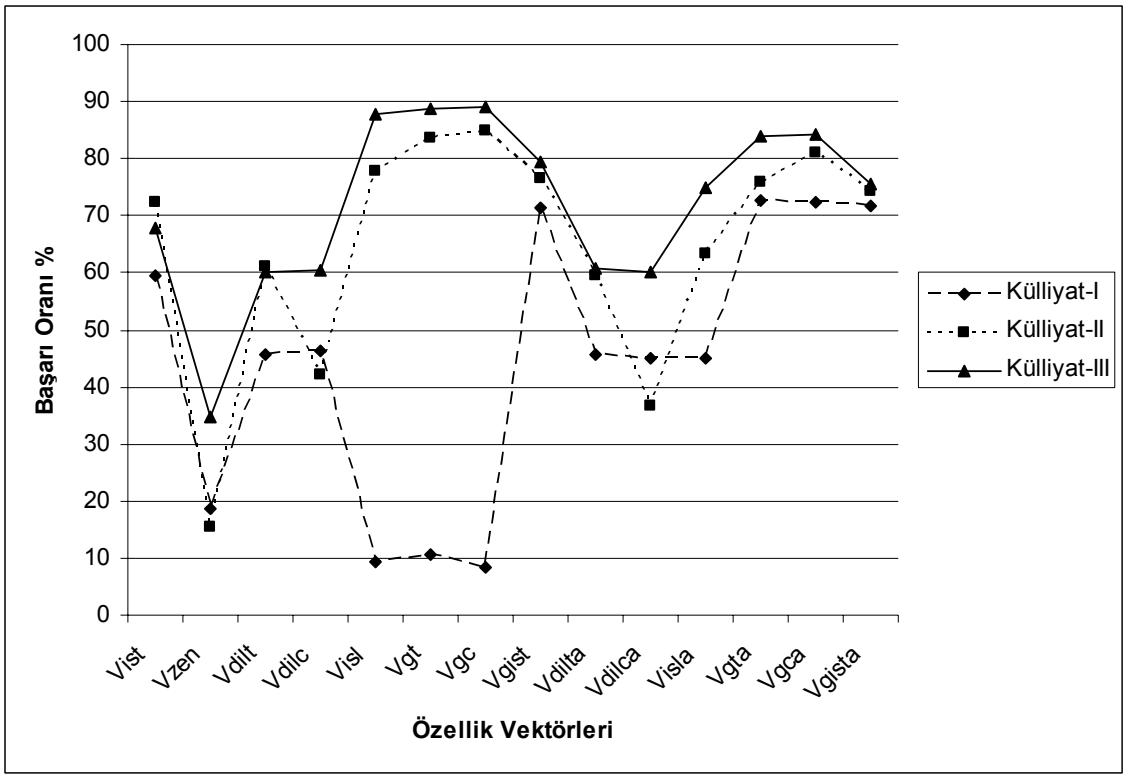
Sınıflandırma yapmak amacıyla Çok Katmanlı Algılayıcı yöntemi ile deney yapıldığında külliyatlardan elde edilen başarı oranları Şekil 5.4'deki grafikte gösterilmiştir.

Çok Katmanlı Algılayıcı yönteminin, Külliyat-I üzerindeki en yüksek başarısı %72,7 olup, Vgta özellik vektöründen elde edilmiştir. En kötü sonuç ise, Vgc özellik vektörü ile alınan %8,5'tir. Külliyat-II ile yapılan deneylerde bu yöntemin en iyi performansı Vgc ile alınan %85'tir. En kötü performans ise, Vzen özellik vektörü ile alınan % 15,5'tir. Külliyat-III'te alınan en yüksek başarı %89,2 olup, Vgc özellik vektöründen elde edilmiştir. En kötü sonuç Vzen özellik vektöründen elde edilen %34,6'dır. ÇKA yöntemi Külliyat-I'de diğer külliyatlara nazaran daha kötü bir performans göstermiştir. Külliyat-I için Visl, Vgt ve Vgc ile alınan sonuçlar diğer külliyatlara göre oldukça düşüktür.

Özellik vektörlerine göre ortalama alındığında bu yöntemin en başarılı olduğu özellik vektörü %77,5 başarı ile Vgta'dır. Sınıflandırma için en elverişsiz özellik vektörü ise %23 ile Vzen'dir.

Her külliyat için başarı ortalamaları hesaplandığında Külliyat-III'ün, %72 başarı oranı ile Çok Katmanlı Algılayıcı'nın en başarılı olduğu külliyat olduğu gözlemlenmiştir.

Genel olarak üç külliyat için Çok Katmanlı Algılayıcı yönteminin ortalama başarısı %60,3'tür.



Şekil 5.4 Çok katmanlı algılayıcı yöntemi ile deney yapıldığında külliyatlardan elde edilen başarı oranları

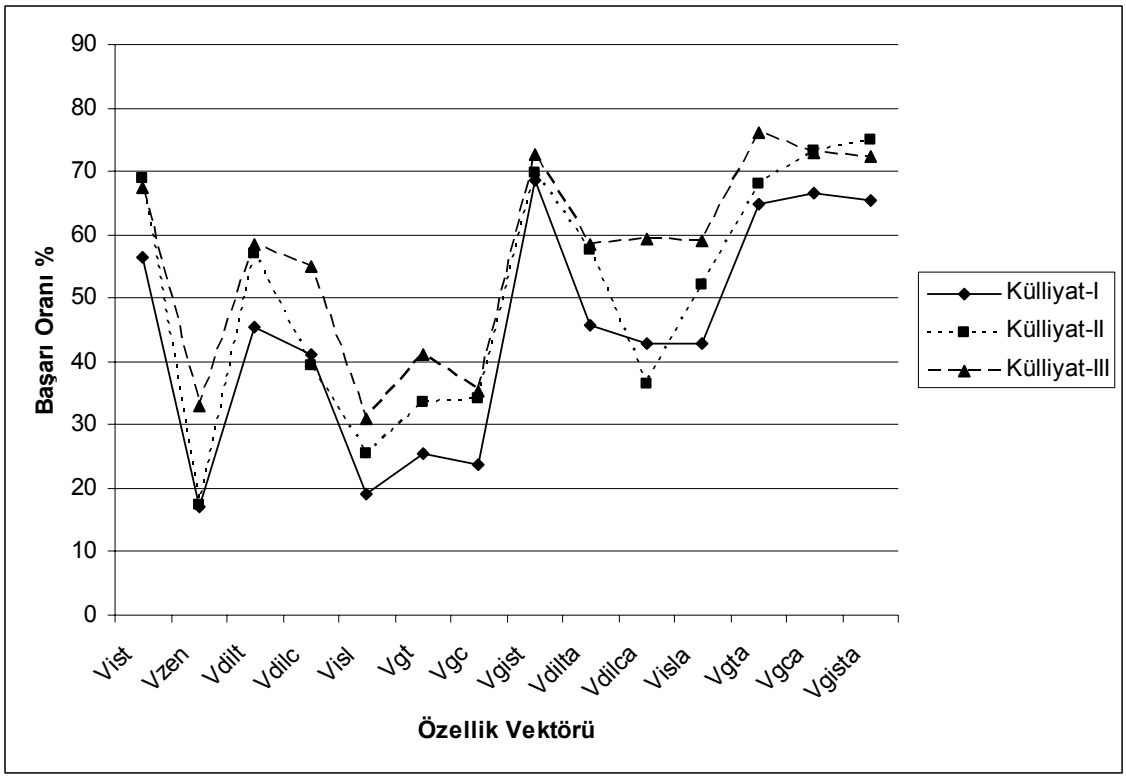
5.2.5 K-Enyakın Komşuluk

K-Enyakın Komşuluk ile sınıflandırma yapılarak her üç külliyattan elde edilen başarı oranları Şekil 5.5'deki grafikte gösterilmiştir.

K-Enyakın Komşuluk yönteminin Külliyat-I üzerindeki en yüksek başarısı %68,6 olup, Vgist özellik vektöründen elde edilmiştir. En kötü sonuç ise, Vzen özellik vektörü ile alınan %17,1'dir. Külliyat-II ile yapılan deneylerde bu yöntemin en iyi performansı Vgista ile alınan %74,9 sonucudur. En kötü performans ise, Vzen özellik vektörü ile alınan %17,5'tir. Külliyat-III'te alınan en yüksek başarı %76,2 olup, Vgta özellik vektöründen elde edilmiştir. En kötü sonuç yine Visl özellik vektöründen elde edilen %31,1'dir.

Özellik vektörlerine göre ortalama alındığında K-Enyakın Komşuluk yönteminin en başarılı olduğu özellik vektörü %71 başarı ile Vgca'dır. Sınıflandırma için en elverişsiz özellik vektörü ise %22,5 ile Vzen'dir.

Her külliyat için başarı ortalamaları hesaplandığında %56,6'lık başarı oranıyla Külliyat-III, K-Enyakın Komşuluk'un en başarılı olduğu külliyattır.



Şekil 5.5 K-Enyakın komşuluk ile sınıflandırma yapılarak her üç külliyattan elde edilen başarı oranları

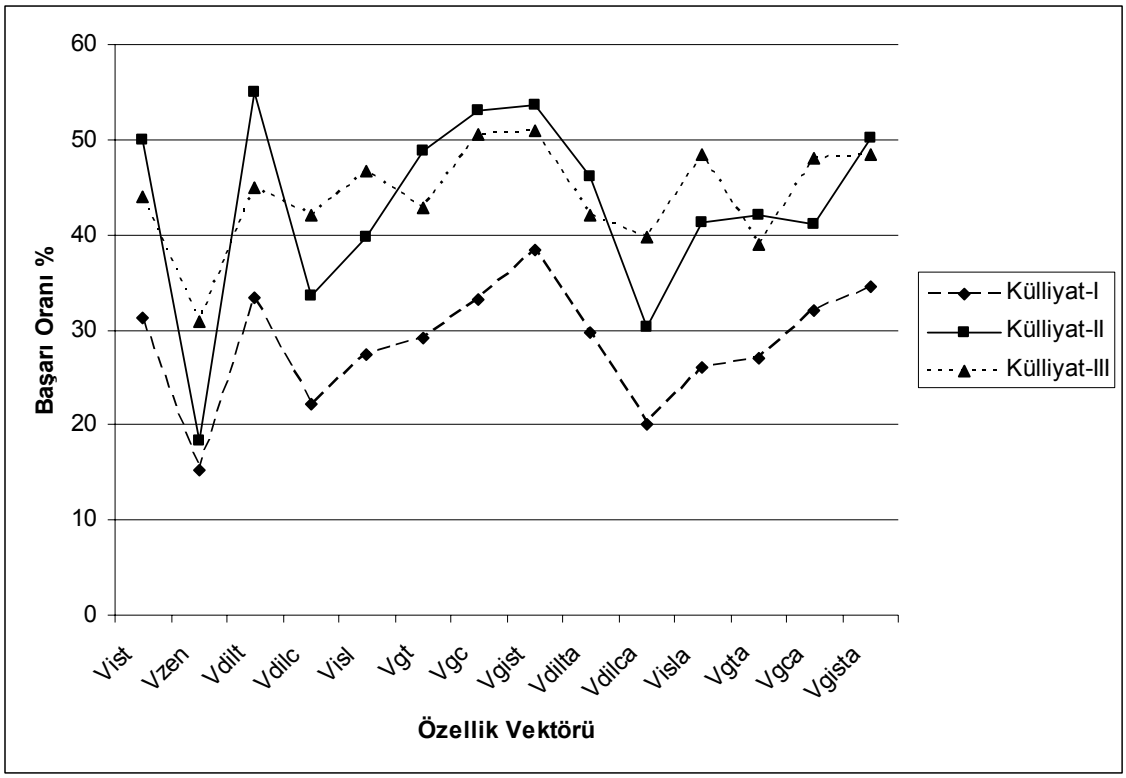
Genel olarak üç külliyat için K-Enyakın Komşuluk yönteminin ortalama başarısı % 50,6'dır.

5.2.6 Öz Düzenleyici Özellik Haritası

Öz Düzenleyici Özellik Haritası yöntemi ile her üç külliyattan elde edilen başarı oranları Şekil 5.6'deki grafikte gösterilmiştir.

Öz Düzenleyici Özellik Haritası yönteminin Külliyat-I üzerindeki en yüksek başarısı % 38,3 olup, Vgist özellik vektöründen elde edilmiştir. En kötü sonuç ise, Vzen özellik vektörü ile alınan %15,2'dir. Külliyat-II ile yapılan deneylerde bu yöntemin en iyi performansı Vgc ile alınan %53 sonucudur. En kötü performans ise, Vzen özellik vektörü ile alınan %18,4'tür. Külliyat-III'te alınan en yüksek başarı %51 olup Vgist özellik vektöründen elde edilmiştir. En kötü sonuç yine Vzen özellik vektöründen elde edilen %30,9'dur.

Özellik vektörlerine göre ortalama alındığında Öz Düzenleyici Özellik Haritası'nın en başarılı olduğu özellik vektörü %47,6 başarı ile Vgist'tir. Sınıflandırma için en elverişsiz özellik vektörü ise %21,5 ile Vzen'dir.



Şekil 5.6 Öz düzenleyici özellik haritası yöntemi ile her üç külliyattan elde edilen başarı oranları

Her külliyat için başarı ortalamaları hesaplandığında Külliyat-III'ün, %44,2 başarı oranı ile Öz Düzenleyici Özellik Haritası'nın en başarılı olduğu külliyat olduğu gözlemlenmiştir.

Genel olarak üç külliyat için Öz Düzenleyici Özellik Haritası yönteminin ortalama başarısı %38,6'dır.

5.3 Sınıflandırıcı Birleştirmeye Göre Deneysel Sonuçlar

Sadece bir sınıflandırıcıya bağlı kalmamak, başarıyı arttırmak düşüncesiyle külliyatlar üzerinde yapılan deneyler sonucunda başarılı performans gösteren üç sınıflandırıcı seçilerek oylama mantığıyla birleştirme yapılmıştır. Bu yöntemler Naive Bayes, Destek Vektör Makinesi ve Rastgele Orman'dır. Birleştirilmiş sınıflandırıcılar daha önceki deneylerde en iyi sonuçları veren özellik vektörleri üzerinde denenmiştir. Bu özellik vektörleri Vist, Visl, Vgt, Vgta, Vgc, Vgca, Vgist'tir.

Külliyat-I üzerinde birleştirilmiş sınıflandırıcı ile yapılan çalışmada elde edilen sonuçlar Çizelge 5.4' te verilmiştir.

Çizelge 5.4 Külliyat-I için sınıflandırıcı birleştirme sonuçları

	Sınıflandırıcılar		
Özellik Vektörü	**NB-RO %**	**NB-DVM %**	**NB-DVM-RO %**
Vist	62,2	55,7	62,7
Visl	63	62,9	63
Vgt	68	67,7	68
Vgta	74,8	74	74,8
Vgc	66,5	66,5	66,5
Vgca	76,5	75,2	76,5
Vgist	71,1	68	71,1
Ort. Başarı	68,9	67,1	69

Külliyat-I'de sınıflandırıcı birleştirme ile alınan sonuçlar genel olarak bireysel sınıflandırıcılarla elde edilenden daha iyidir. Ancak, sadece DVM kullanıldığında en yüksek başarı %82,1 elde edilmişken, sınıflandırıcıların birleştirilmesinde seçilen yöntemlerden biri DVM olmasına rağmen en yüksek sonuç olarak %76,5 alınmıştır. Bunun sebebi DVM ile birleştirilen diğer sınıflandırıcıların DVM'den daha farklı tahmin değerleri vermesidir. Sınıflandırma işlemi, oylama mantığına dayandığından bu da başarının düşmesine sebep olmuştur.

Birleştiricilerle elde edilen en yüksek başarı Vgca vektörü ile NB, DVM ve RO yöntemlerinin birleştirilmesinden elde edilen %76,5'tir. En düşük başarı ise Vist vektöründeki özelliklerle NB ve DVM yöntemleri birleştirilerek yapılan sınıflandırmada alınan %55,7'dir.

Birleştirilmiş sınıflandırıcıların ortalamaları alındığında en başarılı kombinasyonun %69 ile NB, DVM ve RO olduğu gözlemlenmiştir. Özellik vektörleri bazında ortalama hesaplandığında ise Külliyat-I için bu birleştirmelerle en iyi sonuç veren özellik vektörü % 76'lık başarı oranı ile Vgca'dır.

Külliyat-I'in sınıflandırıcı birleştirmedeki genel başarısı %68,3'tür.

Külliyat-II üzerinde birleştirilmiş sınıflandırıcı ile yapılan çalışmada elde edilen sonuçlar Çizelge 5.5'te verilmiştir.

Çizelge 5.5 Külliyat-II için sınıflandırıcı birleştirme sonuçları

	Sınıflandırıcılar		
Özellik Vektörü	**NB-RO %**	**NB-DVM %**	**NB-DVM-RO %**
Vist	74,3	72,4	74,3
Visl	59	58,7	59
Vgt	65,4	65,4	65,4
Vgta	81,2	80,3	82
Vgc	65,7	65,7	65,7
Vgca	84,1	84,1	83,8
Vgist	78,4	76,8	78,4
Ort. Başarı	72,6	71,9	72,7

Külliyat-II'de alınan sonuçların bir kısmı bireysel sınıflandırıcılarla elde edilenden daha iyi iken, bir kısmı daha başarısızdır.

Sınıflandırıcıların birleştirilmesinde elde edilen en yüksek başarı %84,1 ile Vgca vektörü kullanıldığında NB ve RO yöntemlerinin birleştirilmesiyle alınmıştır. En düşük performans ise Visl vektöründeki özelliklerle NB ve DVM yöntemleri birleştirilerek yapılan sınıflandırmada elde edilen %58,7'lik sonuçtur.

Birleştirilmiş sınıflandırıcıların ortalamaları hesaplandığında en başarılı kombinasyonun % 72,7 ile NB, DVM ve RO olduğu gözlemlenmiştir. Özellik vektörlerinin en iyi sonuç vereni ise % 84'lük başarı oranı ile Vgca'dır.

Külliyat-II'in sınıflandırıcı birleştirmedeki genel başarısı %72,4'tür.

Külliyat-III üzerinde birleştirilmiş sınıflandırıcı ile yapılan çalışmada elde edilen sonuçlar Çizelge 5.6'da verilmiştir.

Diğer külliyatlarda olduğu gibi Külliyat-III'te de, elde edilen başarı oranlarının bir kısmı bireysel sınıflandırıcılarla elde edilenden daha iyi iken, bir kısmı daha kötüdür.

Külliyat-III'teki sonuçlar arasında en yüksek başarı %86,3 olup, Vgta vektörü ile NB, DVM ve RO yöntemlerinin birleştirilmesiyle alınmıştır. En düşük performans ise Vgc vektöründeki özelliklerle NB, RO ve DVM yöntemleri birleştirilerek yapılan sınıflandırmada elde edilen % 66,5'tir.

Çizelge 5.6 Külliyat-III için sınıflandırıcı birleştirme sonuçları

	Sınıflandırıcılar		
Özellik Vektörü	**NB-RO %**	**NB-DVM %**	**NB-DVM-RO %**
Vist	76,2	72	75,8
Visl	76,2	76,2	76,2
Vgt	79	79	79
Vgta	85,7	85	86,3
Vgc	78,4	78,4	66,5
Vgca	85,7	85	86
Vgist	78,7	76,2	78,7
Ort. Başarı	80	78,8	78,4

Birleştirilmiş sınıflandırıcıların verdikleri başarı oranlarının ortalamaları hesaplandığında en başarılı birleşimin %80 ile NB ve RO olduğu gözlemlenmiştir. %85'lik başarı oranı ile özellik vektörlerinden en iyi sonuç vereni Vgta olmuştur.

Külliyat-III'ün sınıflandırıcı birleştirmedeki genel başarısı %79'dur.

6. SONUÇ

Bu çalışmanın amacı, yazarı bilinmeyen Türkçe dokümanların yazarlarının belirlenmesidir. Yazar tanıma yapılırken 18 yazara ait 35 adet dokümandan oluşan üç farklı külliyat üzerinde deneyler yapılmıştır. Bu külliyatlara farklı özellik vektörleri uygulanarak her doküman için bir özellik vektörü oluşturulmuştur. Bu vektörler istatistiksel, dilbilgisel, kelime zenginliğine dayalı ve işlevsel kelimelerin frekanslarından oluşan 5 ana özellik vektörüdür. Ayrıca, seçilen özellik azaltıcı bir yöntem ile bu vektörlerin ayırt edici özellikleri seçilerek alt vektörler oluşturulup vektör sayısı 14'e çıkarılmıştır.

Sınıflandırma yapılırken Rastgele Orman, Naive Bayes, Destek Vektör Makinesi, Çok Katmanlı Algılayıcı, K-Enyakın Komşuluk ve Öz Düzenleyici Özellik Haritası yöntemleri uygulanmıştır. Bu yöntemlerden alınan en başarılı sonuçlar ve bu sonuçları veren özellik vektörleri Çizelge 6.1'de gösterilmektedir.

Çizelge 6.1 Yöntemlerden alınan en başarılı sonuçlar ve bu sonuçları veren özellik vektörleri

Sınıflandırıcı	**Özellik Vektörü ve Başarısı**	**Ortalama Başarı**
RO	Vgist - % 81	% 65
NB	Vgta - % 85	% 68,5
ÇKA	Vgc - % 89,2	% 72
DVM	Vgt - % 87,6	% 64
K-NN	Vgta - % 76,2	% 56,6
ÖDH	Vgist - % 51	% 44,2

Buna göre külliyatlar arasında en başarılı olan Külliyat-III'tür. Bunun sebebi, Külliyat-III'deki yazar sayısının diğerlerine göre daha az olması ve sadece aynı konuda yazan yazarlardan oluşmasıdır. Külliyat-I, 18 yazardan oluşmakta ve kullanılan dokümanlar farklı konularda seçilmiş olduğundan en düşük performansı vermiştir. Külliyat-II'de yine Külliyat-I'de olduğu gibi farklı konulardan oluşmakta, ancak yazar sayısı 18 yerine 9 olduğu için Külliyat-I'e göre sınıflandırma başarısı daha yüksektir. Külliyat-III'de ise hem yazar sayısı 9'dur, hem de aynı konuda yazan yazarlardan oluştuğu için diğer külliyatlara göre çok daha başarılıdır. Buradan şu rahatlıkla söylenebilir ki, bir yazar tanıma sisteminin başarısı yazar sayısına ve dokümanların yazıldığı konu çeşitliliğine bağlıdır.

En iyi sonuçları veren özellik vektörleri Vgt, Vgta, Vgc ve Vgist'tir. Vgt ve Vgc istatistiksel, dilbilgisel, kelime zenginliğine dayalı ve işlevsel kelimelerin frekanslarından oluşan vektörlerdir. En yüksek performanslar bu vektörlerden alınmıştır. Bu özellikler yazar tanıma

için oldukça ayırt edici özelliklerdir. Bu özellikler ayrı ayrı vektörler halinde külliyatlara uygulandıklarında en başarılı olan vektör, istatistiksel özelliklerden oluşanıdır. En kötü performansı ise kelime zenginliğine dayalı olan özellik vektörü göstermiştir.

NB, RO ve DVM yöntemleri birleştirilerek oluşturulan melez sistemlerden alınan sonuçlar, herbir sınıflandırıcının en başarılı olduğu olduğu külliyat ve özellik vektörleri Çizelge 6.2'de gösterilmektedir.

Çizelge 6.2 Sınıflandırıcı birleştirilerek elde edilen en iyi sonuçlar, külliyat ve özellik vektörleri

Birleştirilen Sınıflandırıcılar	**Külliyat**	**Özellik Vektörü ve Başarısı**	**Ortalama Başarı**
NB + RO	III	Vgta, Vgca - % 85,7	% 80
NB + DVM	III	Vgta, Vgca - % 85	% 78,8
NB + RO + DVM	III	Vgta - % 86,3	% 78,4

Oluşturulan melez sistemler sonucunda da en iyi performans yine Külliyat-III'ten elde edilmiştir. NB, RO ve DVM yöntemlerinin birleşiminden alınan % 86,3'lük bir başarı alınmıştır. İstatistiksel, dilbilgisel, kelime zenginliğine dayalı ve işlevsel kelimelerin frekanslarından oluşan vektörlerden özellik azaltma yoluyla elde edilen Vgta ve Vgca vektörleri ise en başarılı vektörlerdir.

Yapılan deneyler sonucunda, sınıflandırma yöntemlerinin birleştirilip kullanılmasındansa bireysel olarak uygulandığında daha iyi sonuçlar verdiği gözlemlenmiştir. Bu külliyatlar üzerinde sınıflandırıcı birleştirmenin çok efektif bir yöntem olmadığı söylenebilir.

Bu çalışmada yer verilmeyen n-gram gibi kullanılmayan başka özellikler ileriki çalışmalarda ele alınarak yazar tanıma için daha da kapsamlı bir analiz ve genişletilmiş bir özellik uzayıyla daha başarılı sınıflandırmalar yapılabilir.

Kısaca bu çalışma, bugüne kadar Türkçe dokümanların yazarlarının belirlenmesinde ve çıkarılan özellik vektörlerinin zenginliği açısından en kapsamlı çalışmadır.

KAYNAKLAR

Aha, D. ve Kibler, D., (1991), "Instance-based learning algorithms", Machine Learning, vol.6, pp. 37-66.

Argamon, S. ve Levitan, S., (2005), Measuring the usefulness of function words for authorship attribution. In Proceedings of the 2005 ACH/ALLC Conference, Victoria, BC, Canada.

Baayen, H., Halteren, H. V., Neijt, A. ve Tweedie, F., (2002), An experiment in authorship attribution. 6th JADT.

Baoli, L., Yuzhong, C. ve Shiwen, Y., (2002), A Comparative Study on Automatic Categorization Methods for Chinese Search Engine [A]. In: Proceedings of the Eighth Joint International Computer Conference [C]. Hangzhou: Zhejiang University Press, 117-120.

Binongo, J.N.G., (2003), Who wrote the 15th book of oz? an application of multivariate statistics to authorship attribution. Computational Linguistics, 16(2):917.

Breiman, L., (2001), Random forests. Machine Learning Journal, 45:5–32.

Burrows, J., (1987), Word patterns and story shapes: the statistical analysis of narrative style. Literary and linguistic Computing, 2:61-70

Burrows, J.F., (1992), Computers and the Study of Literature. In: C. S. Butler (Ed.), Computers and Written Texts. Oxford: Blackwell. (pp. 167-204)

Chakrabarti, S., Dom, B., Agrawal, R. ve Raghavan, P., (1997), Using taxonomy, discriminants and signatures for navigating in text databases. In Proceedings of the 23rd VLDB Conference.

de Vel, O., Anderson, A., Corney, M., Mohay, G., (2001), Multitopic e-Mail Authorship Attribution Forensics; ACM Conference on Computer Security – Workshop on Data Mining for Security Applications.

de Vel, O., Anderson, A., Corney, M., Mohay, G., (2002), Language and Gender Author Cohort Analysis of E-mail for Computer Forensics. Proceedings Digital Forensic Research Workshop.

de Vel, O., (1999), Evaluation of Text Document Categorisation Techniques for Computer Forensics. Journal of Computer Security.

Diederich, J., Kindermann, J., Leopold, E., Paass, G., (2000), Authorship attribution with support vector machines Poster presented at The Learning Workshop

Diri, B., Amasyalı, M.F., (2003), "Automatic Author Detection for Turkish Text", 13th International Conference on Neural Information Processing, Turkey.

Domingos, P. ve Pazzani, M., (1997), On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning, 29:103-130.

Domingos, P. ve Pazzani, M., (1996), Beyond Independence: conditions for the optimality of the simple bayesian classifier. Thirteenth International Conference on Machine Learning(ICML).

Dumais, S., Platt, J., Heckerman, D., Sahami, M., (1998a), Inductive Learning Algorithms and Representations for Text Categorization 7th International Conference on Information and Knowledge Management, pp. 148-152

Dumains, S., Platt, J., Hackerman, D. ve Sahami, M., (1998b) Inductive learning algorithms and representations for text classification. Seventh International Conference on Information and Knowledge Management.

Friedman, N., Geiger, D., Goldszmidt, M., (1997), Bayesian network classifiers. Machine Learning,29:131-163.

Frietag, D. ve McCallum, A., (1999) Information extraction with hmms and shrinkage. In Proceedings of the AAAI'99 Workshop on Machine Learning for Information Extraction.

Fung, G., Mangasarian, O., (2003), The Disputed Federalist Papers: SVM Feature Selection via Concave Minimization Proceedings of the 2003 Conference of Diversity in Computing. Atlanta, Georgia, USA, 42-46.

Gerritsen, C.M., (2003), Authorship Attribution Using Lexical Attraction, Master Thesis Department of Electrical Engineering and Computer Science, MIT

Graham, N., Hirst, G., Bhaskara, M., (2003) Segmenting a document by stylistic character 18th International Joint Conference on Artificial Intelligence

Hall, M.A., (1998), Correlation-based feature selection machine learning. PhD thesis, Department of Computer Science, University of Waikato, New Zealand.

Hall, M.A., Smith, L.A., (1997), Feature subset selection: a correlation based filter approach, Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems. Springer. 855-858.

Holmes, D.I., (1994), Authorship Attribution. Computers and the Humanities 28(2): 87-106.

Holmes, D.I. ve Forsyth, R.S., (1995), The Federalist Revisited: New Directions in Authorship Attribution. Literary and Linguistic Computing 10(2): 111-127.

Holmes, D.I., Robertson, M. ve Paez, R., (2001), Stephen crane and the new-york tribune: A case study in traditional and non-traditional authorship attribution. Computers and the Humanities, 35(3):315-331.

Joachims, T., (1998), Text Categorization with Support Vector Machines: Learning witth Many Relevant Features 10th European Conference on Machine Learning, pages 137-142, Heidelberg, Germany.

John, G.H. ve Langley, P., (1995), Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. pp. 338-345. Morgan Kaufmann, San Mateo.

Juola, P. ve Baayen, H., (2003), A controlled-corpus experiment in authorship identification by cross-entropy. Literary and Linguistic Computing.

Kelih, E., Antic, G., Grzybek, P., Stadlober, E., (2005), Classification of Author and/or Genre? The Impact of Word Length. In: C. Weihs and W. Gaul (Eds.), Classification – The Ubiquitous Challenge. Springer, Heidelberg;498–505.

Kepenekci, B., Akar, G.B., (2004), "Destekçi vektör makinası ile yüz sınıflandırma (Face

classification with support vector machine)", IEEE 12.Sinyal işleme ve iletişim uygulamaları kurultayı, 28-30 Nisan, Kuşadası, Türkiye

Khmelev, D., Tweedie, F., (2001), Using Markov Chains for Identification of Writers Literary and Linguistic Computing, 16(4), 299—307

Kohonen T, Kaski S, Lagus K, Salojarvi J, Honkela J, Paatero V, Saarela A (2000) Self organization of a massive document collection. IEEE Transactions on Neural Network vol.11, No. 3

Koppel, M., Schler, J., (2003), Exploiting Sty list ic Idiosyncrasies for Authorship Attribution IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis

Koppel, M., Schler, J., (2004), Authorship Verification as a One-Class Classification Problem, in Proceedings of 21st International Conference on Machine Learning, pp. 489-495.

Manning, C.D. ve Schutze, H., (1999), Foundations of Statistical Natural Language Processing [M]. Cambridge: MIT Press.

McCallum, A., Nigam, K., (1998), A Comparison of Event Models for Naive Bayes Text Classification. AAAI-98 Workshop on Learning for Text Categorization.

Mitchell, T.M., (1997), Machine Learning. syf:177-184. Mcgraw-Hill Companies.

Náther, P., (2005), N-gram based Text Categorization, Diploma Thesis, Comenius University

Öztemel, E., (2003), Yapay Sinir Ağları. Papatya Yayıncılık

Pal, M., ve Watanachaturaporn, P., (2004), Support vector machines. In P. K. Varshney & M. K. Arora (Eds.), Advanced image processing techniques for remotely sensed hyperspectral data: Springer-Verlag.

Peng, F., Schuurmans, D., Keselj, V., Wang, S., (2003), Language Independent Authorship Attribution using Character Level Language Models. EACL 267-274

Peng, F., Schuurmans, D., Combining Naive Bayes and n-Gram Language Models for Text Classification, ECIR 335-350

Peng, F., Keselj, V., Cercone, N., Thomas, C., (2003), N-Gram-based Author Profiles for Authorship Attribution. PACLING'03, Dalhousie University, Halifax, Nova Scotia, Canada.

Platt, J., (1998), "Fast Training of Support Vector Machines using Sequential Minimal Optimization". Advances in Kernel Methods - Support Vector Learning, B. Scholkopf, C. Burges, and A. Smola, eds., MIT Press

Rumelhart, D.E., Hinton, G.E. ve Williams, R.J., (1986), "Learning Internal Representations by Error Propagation" pp. 318-362 in Parallel Distributed Processing, ed. Rumelhart, D.E. and McClelland, J.L. (Eds), MIT Press

Salton, G., Yang, C. ve Wong, A., (1975), “A vector-space model for automatic indexing”, Communications of the ACM, Vol. 18, No. 11, pp. 613–620

Sang-Bum, K., Hee-Cheol S. ve Hae-Chang R., (2003), Poisson Naive Bayes for Text Classification.>=2000, 6th IRAL , Sapporo, Japan

Stamatatos, E., Fakotakis, N., Kokkinakis, G., (1999), Automatic Authorship Attribution. EACL, 1999

Stamatatos, E., Fakotakis, N., Kokkinakis, G., (2000), Automatic Text Categorization in Terms of Genre and Author, Computational Linguistics, pp.471-495.

Vladimir, N.V., (1995), The Nature of Statistical Learning Theory. http://automotif.bioinfo.pl/method.htm

Wolters, M. ve Kirsten M., (1999), "Exploring the use of linguistic features in domain and genre classification", Proceedings of EACL99.

Yang, Y. ve Liu X., (1999), A Re-examination of Text Categorization Methods [A]. In: Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval [C]. 42-49.

Zhao, Y., Zobel, J., (2005), Effective and Scalable Authorship Attribution Using Function Words, RMIT University,Melbourne, Australia.

İNTERNET KAYNAKLARI

[1] www.cs.waikato.ac.nz/ml/weka/

[2] www.hurriyet.com.tr

[3] www.vatanim.com.tr

[4] www.sabah.com.tr

[5] http://automotif.bioinfo.pl/method.htm

[6] http://inpc55.et.tudelft.nl/research/papers/thesis/node16.html

[7] http://www.stat.berkeley.edu/ users/breiman/ RandomForests/cc_home.htm

[8] http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-12.html

[9] www.tdk.org.tr

ÖZGEÇMİŞ

Doğum tarihi	05.03.1980	
Doğum yeri	Malatya	
Lise	1995–1998	Bartın Anadolu Lisesi
Lisans	1999–2003	Yıldız Üniversitesi Elektrik-Elektronik Fak. Bilgisayar Mühendisliği Bölümü
Yüksek Lisans	2003–2006	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği

Çalıştığı kurumlar

2004–2004	YTÜ Fen Bilimleri Enstitüsü, Araştırma Görevlisi.
2004-Devam ediyor	TURKCELL İletişim Hiz. A.Ş., Yazılım Mühendisi