KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

METİN MADENCİLİĞİ YÖNTEMLERİ İLE YAZAR TANIMA: DİVAN EDEBİYATI ÖRNEĞİ

YÜKSEK LİSANS TEZİ

Ali Osman BİLGİN

HAZİRAN 2018
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

METİN MADENCİLİĞİ YÖNTEMLERİ İLE YAZAR TANIMA: DİVAN EDEBİYATI ÖRNEĞİ

Ali Osman BİLGİN

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"YÜKSEK LİSANS (İSTATİSTİK)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 29 / 05 /2018
Tezin Savunma Tarihi : 26 / 06 /2018

Tez Danışmanı : Dr. Öğr. Üyesi Tolga BERBER

Trabzon 2018

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İstatistik ve Bilgisayar Bilimleri Anabilim Dalında
Ali Osman BİLGİN Tarafından Hazırlanan

METİN MADENCİLİĞİ YÖNTEMLERİ İLE YAZAR TANIMA: DİVAN EDEBİYATI ÖRNEĞİ

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 29 / 05 / 2018 gün ve 1755 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Asiye Mevhibe COŞAR .....

Üye : Doç. Dr. Kamil ALAKUŞ .....

Üye : Dr. Öğr. Üyesi Tolga BERBER .....

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

"Metin Madenciliği Yöntemleri ile Yazar Tanıma: Divan Edebiyatı Örneği" isimli bu tez çalışması, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstatistik ve Bilgisayar Bilimleri Anabilim Dalı, Yüksek Lisans Programı'nda hazırlanmıştır.

Tez çalışma süresince değerli yardım ve katkılarıyla beni yönlendiren danışman hocam sayın Dr. Öğr. Üyesi Tolga BERBER'e, yüksek lisans eğitimi yapma fırsatı bulduğum İstatistik ve Bilgisayar Bilimleri Bölümündeki tüm hocalarıma, Divan Edebiyatı ile kaynaklara erişim ve kelimelerdeki yazım hatalarının düzeltilmesi konusunda yardımcı olan sayın Prof. Dr. A. Mevhibe COŞAR'a ve Divan Edebiyatına ait metin bankası verilerini paylaşan Prof. Dr. A. Atilla ŞENTÜRK'e teşekkürü borç bilirim.

Eğitim ve öğretim hayatım boyunca maddi ve manevi her türlü desteği sağlayan annem, babam ve ağabeyime çok müteşekkir olduğumu belirtmek isterim. Son olarak hayatıma anlam katan ve yeni doğan kızımın annesi olan eşim Özge YILDIZ BİLGİN'e bu zor süreçte bana her türlü desteği verdiği için en içten teşekkürlerimi sunarım.

Ali Osman BİLGİN

Trabzon 2018

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Metin Madenciliği Yöntemleri ile Yazar Tanıma: Divan Edebiyatı Örneği” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Dr. Tolga BERBER’in sorumluluğunda tamamladığımı, verileri kendim topladığımı, analizleri ilgili laboratuvarlarda yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 26/06/2018

Ali Osman BİLGİN

İÇİNDEKİLER

Sayfa No

Yüksek Lisans Tezi

ÖZET

METİN MADENCİLİĞİ YÖNTEMLERİ İLE YAZAR TANIMA: DİVAN EDEBİYATI ÖRNEĞİ

Ali Osman BİLGİN

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik ve Bilgisayar Bilimleri Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Dr. Tolga BERBER
2018, 67 Sayfa, 4 Ek Sayfa

Özellikle 21. yy'ın başından itibaren bilişim teknolojilerinin artan hızda gelişmesi ve gündelik hayatın neredeyse her aşamasına entegre olması ile birçok alanda büyük miktarda veri toplanmaya başlanmıştır. Bu verilerin sistematik bir şekilde depolanması, hızlı bir şekilde yönetilmesi ve kolaylıkla analiz edilebilmesi için veri tabanı yönetim sistemleri kullanılmaktadır. Bilişim dünyasındaki bilgilerin büyük çoğunluğu düz metin, e-posta, resim, ses ve video dosyaları gibi sistematik olmayan verilerdir. Geleneksel istatistiki yöntemler ile analiz edilemeyen bu verilerden anlamlı bilgiler çıkarabilmek için veri madenciliği, metin madenciliği, duygu analizi, görüntü ve ses işleme gibi yöntemler kullanılmaktadır. Bu çalışmada incelenen veriler de metin formatında olduğundan metin madenciliği yöntemleri kullanılmıştır. Metin madenciliğinin temel hedefleri metinlerin konularına göre ayrıştırılması, özetinin çıkarılması, başlıklarının eklenmesi ve yazarlarının belirlenmesidir. Bu çalışma ile 25 divan edebiyatı şairine ait eserlerin yazarlarını belirleyen bir sistem geliştirilmiştir. Metin madenciliğinin metin sınıflandırma algoritmalarından yararlanılarak sözcüklerin analiz edilmesine dayanan bu sistemde her bir parametrenin olası değerleri için 20 farklı model kurulmuştur. Her modelin tek tek karşılaştırılması neticesinde %91,45'lık doğruluk ve %90,23'lük f-değerine ulaşılmıştır. Böyle bir çalışmanın uzun vadede yazarı bilinmeyen eserlerin sahiplerinin tespitine dair tahminleri destekleyebileceği düşünülmektedir.

Anahtar Kelimeler: Metin Madenciliği, Metin Sınıflandırma, Yazar Tanıma

Master Thesis

SUMMARY

AUTHORSHIP RECOGNITION WITH TEXT MINING METHODS: THE EXAMPLE OF DIVAN LITERATURE

Ali Osman BİLGİN

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Statistical and Computer Science Graduate Program
Supervisor: Asst. Prof. Dr. Tolga BERBER
2018, 67 Pages, 4 Pages Appendix

Especially since the beginning of the 21st century, with the increasing speed of information technology and integration into almost every phase of everyday life, a large amount of data has been collected in many areas. Database management systems are used to store such data in order to utilize them systematically, manage them quickly, and analyze them easily. The vast majority of data in the information world is non-systematic, such as pictures and audio files, text files in pdf, word and txt. formats, e-mails, and log files kept on web pages. Methods such as data mining, text mining, sentiment analysis, image and sound processing are used to extract significant information from these data, which cannot be analyzed by traditional statistical methods. For this study text mining methods have been used since the data analyzed in this study are in text format. The main target of text mining is to separate the texts according to their subjects in order to summarize them, to add their titles and to determine their authors. In this study, a system has been developed to determine the authors of 25 poetry works belonging to Divan Literature. In this system, which is based on analyzing the words by using the text classification algorithms of text mining, 20 different models have been established for the possible values of each parameter. As a result of our individual comparisons of each model, 91.26% accuracy and 90.23% f-value ratings were achieved. Such this study is thought to be able to support estimates of the identification of authors of unknown works in the long term.

Key Words: Text Mining, Text Classification, Authorship Recognition

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER DİZİNİ

$tf_{t,d}$: Terim Sıklığı

max_t : En Büyük Terim Sıklığı

N : Toplam Terim Sayısı

df_t : Ters Doküman Sıklığı

Wi : i. Terimin Ağırlığı

u : Eşsiz Eksek Vektörü

Σ : Toplam Sembolü

$\prod$: Çarpım Sembolü

p(b/a) : Koşullu Olasılık

κ : Kappa Katsayısı

BBA : Bağımsız Bileşen Analizi

BR : Basit Regresyon

CSV : Virgülle Ayrılmış Dosyalar

ÇDR : Çoklu Doğrusal Regresyon

ÇKA : Çok Katmanlı Algılayıcı

ÇNB : Çokterimli Naive Bayes

ÇR : Çoklu Regresyon

DA : Diskriminant Analizi

DN : Doğru Negatif

DP : Doğru Pozitif

DR : Doğrusal Regresyon

DVM : Destek Vektör Makinesi

KA : Karar Ağacı

KBÖ : Kosinüs Benzerlik Ölçütü

KÇ : Kelime Çantası

KEK : K En Yakın Komşu

KTÖS : Korelasyon Tabanlı Özellik Seçici

LSA : Latent Semantik Analiz

NB : Naive Bayes

ÖÖH : Öz Düzenleyici Özellik Haritası

ÖUÖ : Öklid Uzaklık Ölçütü

PDF : Taşınabilir Belge Biçimi
RO : Rastgele Orman
SÖÇ : Soyut Özellik Çıkarım
TBA : Temel Bileşen Analizi
TDS : Ters Doküman Sıklığı
TS : Terim Sıklığı
VU : Vektör Uzayı
XML : Genişletilebilir İşaretleme Dili
YHT : Yazım Hatalarının Tespiti
YN : Yanlış Negatif
YP : Yanlış Pozitif
YSA : Yapay Sinir Ağları

1. GENEL BİLGİLER

1.1. Giriş

Tarih öncesi çağlarda kayalara işlenen figürlerle başlayıp matbaanın icadı ile kâğıt üzerinde büyük sayılara ulaşan veriler, günümüzde akla gelen her alanda dijital ortamda kayıt altına alınmaktadır. Teknolojik alt yapının ilerlemesi ile inanılmaz büyüklüklere ulaşan veriler, veri tabanlarında sistematik bir şekilde tutulabildikleri gibi, resim, ses, video ve düz metin gibi yapısal olmayan şekillerde de bulunabilmektedir. Lnych'e göre dijital dünyadaki verilerin %85-90'ı yapısal olmayan verilerden oluşmaktadır [1]. Geleneksel istatistik yöntemlerle analiz edilemeyen bu verilerin analizi için veri madenciliği, metin madenciliği, duygu analizi, görüntü ve ses işleme gibi yöntemler kullanılmaktadır.

Bu tez çalışması kapsamında incelenecek olan metin madenciliği yöntemleri ile metinlerin konularına göre ayrıştırılması, özetinin çıkarılması, başlıklarının eklenmesi, yazarlarının belirlenmesi gibi konularda çalışmalar gerçekleştirilir [2]. Söz konusu çalışmaları gerçekleştirebilmek için bilgi geri getirimi, bilgi çıkarımı, hece analizi, kelime frekans dağılımı, örüntü tanıma, veri madenciliği ve veri görselleştirme gibi yöntemler kullanılır [3]. Bu çalışma kapsamında, divan edebiyatı şiirleri için metin madenciliği yöntemlerini kullanarak bir yazar tanıma sistemi geliştirilmiş ve başarısı değerlendirilmiştir. Bu amaçla, metin madenciliği yöntemlerinin metin sınıflandırma alanında kullanılan algoritmalarından faydalanılmıştır.

1.2. Tezin Konusu ve Önemi

Çalışma kapsamında, divan edebiyatı eserleri için metin madenciliği yöntemleri kullanılarak, şair tanıma işlemi gerçekleştirilmiştir. Başka bir ifade ile yazarı bilinen eserler kullanılarak bir model kurulup eserlerin yazarlarının tahmin edilebilmesini sağlayan istatistiksel bir yaklaşım geliştirilmiştir.

Divan edebiyatının başından sonuna kadar yüzlerce divan şairi yetişmiştir [4]. Şairler ve eserleri tam olarak bilinememekle beraber bu eserlere dijital ortamda ulaşmak oldukça zahmetli ve zordur. Bu bağlamda yapılan çalışmalar neticesinde farklı dönemlerde ve bölgelerde yaşamış 25 divan şairinin eserlerine ulaşılmış ve bu şairler için bir yazar tanıma çalışması gerçekleştirilmiştir. Sonuç olarak, gerçekleştirilen yazar tanıma sistemi sadece bu

şairler için başarılı çıktı üretebilmektedir. Divan edebiyatındaki bütün şairlere ait bir çıkarım yapamamaktadır.

Divan şiirleri Arapça, Farsça ve bugün için çoğunluğu eski Türkçe kelimeler barındırdığından metin madenciliği açısından çalışılması oldukça zor bir konudur [5]. Çünkü metin madenciliği çalışmaları incelenen metnin dil yapısına ilişkin yöntemler kullanır. Dolayısıyla bu çalışma Türkçe, Arapça ve Farsça söz ve yapı kurallarının hakim olduğu Osmanlı Türkçesi için yapılmış olması ile önceki çalışmalardan farklı bir yere sahiptir.

1.3. Divan Şiirleri

Divan edebiyatı, Türklerin Müslümanlığı kabulünden bir müddet sonra başlayan ve Tanzimat Dönemi'ne (1860) kadar devam eden İslam medeniyetinin etkisinde ortaya konulan bir edebiyat türüdür. Bu dönem Arap ve Fars edebiyatının etkisi altında geçmiştir. Türk şairlerinin hayal gücü ve yaratıcılığı ile Arap ve Fars edebiyatlarının harmanlanması neticesinde kendine özgü kuralları, gelenekleri ve anlayışı olan bir dönem haline gelmiştir. Türk edebiyatı Tanzimat dönemine kadar "şiir" temelli olduğundan divan edebiyatının önemli bir kısmını divan şiirleri oluşturmaktadır. Bu nedenle divan edebiyatı denildiğinde genellikle akla divan şiirleri gelmektedir [6].

Divan şiirinde genelde aşk, sevgi, güzellik, ayrılık, hasret, acı, doğa, tabiat ve özlem gibi konular işlenmiştir. Divan şiirinde biçim içeriğe göre daha ön plândadır. Çünkü şairler hünerlerini biçimsel özellikleri dikkate alarak gösterirler. Bu sebeple divan edebiyatında sözcük seçimi çok önemlidir. Kafiyeye, aruz ölçüsüne, edebî sanatlara uygun sözcükler kullanmak esastır [6].

Divan edebiyatını üslup farklılaşmalarını göz önünde bulundurarak başlıca dört döneme ayırmak mümkündür [4].

1.3.1. Kuruluş Devri

Türkçe kelimelerin daha çok kullanıldığı ve İran edebiyatı etkisinin yavaş yavaş kendini hissettirdiği dönemdir. Bu devir II. Mehmet dönemine kadar (1451) devam eder [4].

1.3.2. Geçiş Devri

Eserlerde kullanılan dilin Osmanlıca özellikler gösterdiği ve şairlerin edebiyatta köklü değişiklikler yaptıkları dönemdir. 1512 yılında Yavuz Sultan Selim dönemine kadar sürer [4].

1.3.3. Klasik Devir

Türk edebiyatının İran edebiyatı etkisinden kısmen de olsa kurtularak artık kendi iç gelişimini tamamlayıp özgün eserlerini vermeye başladığı dönemdir [6]. Yaklaşık bir asır süren en ihtişamlı dönemdir. 1603 yılında I. Ahmet döneminde son bulur [4].

1.3.4. Sebk-i Hindi Devri

17. yüzyıl başlarından 19. yüzyılın ikinci yarısına kadar devam eden bu dönemde bir yandan klasik şiir devam ederken öte yandan Sebk-i Hindi (Hind uslübu) denilen bir akım kendini gösterir. Hindistan'da Babürlü Türk-Hind hükümdarlarının saraylarında gelişerek ortaya çıkmış bir tür olan Sebk-i Hindi'de aşırı süs ve sanata, fikri gizlemeye, uzayıp giden tamlamalara ve ince hayallere önem verilmiştir [4].

1.4. Yazar Tanıma

En iyi yazardan ilkokul mezunu bir insana kadar her insanın yazı yazarken şahsına münhasır bir kalemi vardır. Eserlerdeki bu farklılık paragrafın anlam bütünlüğü, cümleler arası geçişler, cümlelerinin uzunluğu ve yapısı, kelimelerin uzunluğu ve farklılığı, noktalama işaretleri, kullanılan bağlaçlar gibi özellikler ile kendini gösterir. Yazar tanıma yöntemleri, bu ve benzer özellikleri kullanarak yazarı bilinmeyen ya da çelişkili olan dokümanların yazarının tespit edilmesi işlemine verilen addır [7].

Eski zamanda yazılmış birçok eserin yazarı bilinemediğinden literatüre anonim olarak geçmiştir. Yazar tanıma yöntemleriyle bu eserlerin yazarları tespit edilebilecektir. Bunun dışında aynı doküman üzerinde hak iddia eden birden fazla kişi içerisinde dokümanın gerçek yazarının kim olduğunun tespiti, dokümanı yazdığını kabul etmeyen kişinin tespiti,

akademik çalışmalarda intihal yapılıp yapılmadığının belirlenmesi yazar tanıma yöntemleriyle gerçekleştirilen başlıca işlemlerdir [8].

Şiir bir edebi metin olduğundan şair tespitinde de yazar tanıma işlemindeki yöntemler kullanılır. Şair tanıma işleminde, yazar tanıma işlemine ek olarak düz yazılarda olmayan uyak, kafiye, redif, aruz vezni, akrostiş gibi şairlerin karakteristik özelliklerini barındıran anlamlı bilgiler de kullanılabilir.

1.4.1. Literatürde Yapılmış Çalışmalar

Yazar tanıma işlemleri 1963 yılında Mosteller ve Wallace'ın yapmış olduğu doküman sınıflama yöntemi ile başlamıştır. Bu alan için mihenk taşı olarak kabul edilen çalışmada belirli konular için özel sözlükler oluşturularak otomatik indeksleme gerçekleştirilmiştir [9]. Modelleme için Naive Bayes (NA) yöntemi kullanılmıştır. Yazar tanıma alanındaki bir diğer önemli çalışma, 2000 yılında Stamatatos ve arkadaşları tarafından Yunan dili için gerçekleştirilmiştir. Toplam 10 yazarın 20'şer makalesi alınarak gerçekleştirilen çalışmada eserlere sözlük, karakter, sözdizimsel ve anlamsal açıdan bakılmıştır. Bunun için cümle sayısı, kelime sayısı, noktalama işaretlerinin sayısı, tamlama ve tamlamadaki kelime sayısı gibi toplam 22 özellik çıkarılarak özellik vektörü oluşturulmuştur. Özellik vektörü, Çoklu Doğrusal Regresyon (ÇDR) ve Diskriminant Analizi (DA) yöntemleri ile analiz edilerek istatistiksel modeller oluşturulmuş ve %69'lık başarıya ulaşılmıştır [7].

Stamatatos ve arkadaşlarının çalışmasından sonra yazar tanıma ile ilgili birçok farklı dilde sayısız çalışma gerçekleştirilmiştir. Divan Edebiyatı Osmanlıca, Arapça, Farsça ve Türkçe sözcükler içerdiğinden sonraki literatür taramaları söz konusu diller için gerçekleştirilmiştir.

İlk olarak Osmanlı Türkçesi için literatür taranmış olup yalnızca Can ve arkadaşlarının 2011 yılında gerçekleştirdiği çalışmaya ulaşılmıştır [10]. Bu çalışma ile Osmanlı Türkçesi ile yazılmış divan edebiyatı eserlerinde metin sınıflandırma yöntemlerinin başarılı olup olmayacağı araştırılmış, başarılı olması durumunda diğer çalışmalara ışık tutacağı düşünülmüştür. Bu sebeple yazar ve eser sayısı az tutularak 15.yy'dan 19.yy'a kadar her bir asır için 2 eser olmak üzere toplam 10 farklı divan metni incelenmiştir. Destek Vektör Makinesi (DVM, Support Vektor Machine) ve NB yöntemleri ile sınıflandırma işlemi gerçekleştirilmiş ve %92,80'lik başarı oranına ulaşılmıştır. Çalışmada herhangi bir gövdeleme yöntemi kullanılmamıştır. Sınıflandırma işlemi sık kullanılan kelimeler, kelime

grupları ve kelimelerin uzunluklarına bakılarak gerçekleştirilmiştir [10]. Sadece 10 doküman ile DVM ve NB sınıflandırma yöntemlerini kullanıp parametreye bağlı tahmin yapan çalışma bu tez çalışması için ışık tutan niteliktedir.

Arapça için ilk olarak 1987 yılında Gheith ve El-Sadany tarafından morfolojik analizler gerçekleştirilmiştir [11]. Daha sonra 1989 yılında Al-Fedaghi ve Al-Anzi tarafından Arapça kelimelerin köklerini bulan bir algoritma geliştirilmiştir [12]. İlerleyen süreçte Arap alfabesi ve Arap dilinde yazar tanıma ve metin sınıflama için birçok çalışma gerçekleştirilmiştir. Bunlar içerisinde Al-Falahi ve arkadaşlarının Arap şiirlerinde makine öğrenimi ile yazar tanıma çalışması konu itibariyle bu tez çalışmasına benzer özellik taşıdığı için ayrı bir yere sahiptir. Kendi çalışmalarından önce Arap şiirlerinde yazar tanıma ile ilgili herhangi bir çalışma ya da araştırma olmadığından bahsetmişlerdir. Çalışmayı farklı konularda rastgele seçilen 73 şiir ile NB ve DVM yöntemlerini kullanarak %98,63 gibi oldukça yüksek bir başarı oranı ile gerçekleştirmişlerdir. Başarılarının yüksek olmasının nedeni şiirleri yalnızca "meter" ve "rhyme" isminde iki gruba ayırmalarıdır [13].

Farsça için literatürde bulunan ilk metin sınıflandırma çalışması 2004 yılında Mazdak'ın yüksek lisans tezi ile oluşturduğu "FarsiSum" isimli metin özetleme programı olmuştur [14]. Daha sonra 2006 yılında Arabsorkhi ve Feili tarafından NB yöntemi ile Farsça metin sınıflayıcısı oluşturulmuştur [15]. Bu tarihten sonra Farsça için çok sayıda metin sınıflandırma çalışması gerçekleştirilmiş olsa da 2009 yılında Hamidi ve arkadaşlarının Farsça şiirlerde DVM ile otomatik sınıflama çalışması bu tez çalışmasıyla benzer konuya sahip olmasından ötürü önem taşımaktadır. Toplam 8 şair ve 12 farklı türden oluşan 136 şiir için yapılan sınıflandırma sonucunda %91'lik başarıya ulaşılmıştır [16].

Türkçe için metin sınıflandırma çalışmalarına 2003 yılında başlanmıştır. Çatal ve arkadaşları n-gramları kullanarak "NECL" isminde bilgisayar tabanlı yazar tanıma sistemi geliştirmişlerdir [17]. Yine aynı yıl Diri ve Amasyalı, Türkçe dokümanların yazarını ve türünü belirlemek için 22 stil belirleyerek sınıflandırma sistemi gerçekleştirmiştir. Yazarın cümle sayısı, kelime sayısı, bir cümledeki ortalama kelime sayısı, kelime uzunluğu, farklı kelimelerin sayısı, kelime zenginliği, bir cümledeki ortalama isim, fiil, sıfat, zarf, edat, zamir, bağlaç, ünlem sayısı, noktalama işaretlerinin sayısı, eksik cümle sayısının toplam cümle sayısına oranı, devrik cümle sayısının toplam cümle sayısına oranı stil olarak belirlenmiştir. Sekiz farklı yazarın her birine ait yirmi farklı makalesiyle oluşturulan derlem %84'lük başarıya ulaşmıştır [18].

Amasyalı ve Diri, 2006 yılındaki başka bir çalışmalarında ise n-gram yöntemini kullanarak dokümanın yazar, tür ve yazarın cinsiyetini belirleyen daha kapsamlı bir metin sınıflandırma çalışması gerçekleştirmişlerdir. NB, DVM, C4.5 ve Rastgele Orman (RO) sınıflandırma algoritmalarını kullanarak yazar tanımada %83, tür belirlemede %93, cinsiyet belirlemede %96'lık başarı elde etmişlerdir [19]. Aynı yıl Türkoğlu yapmış olduğu yüksek lisans çalışmasında melez yaklaşımlarla yazar tanıma işlemini gerçekleştirmiştir. Dokümanların istatistiksel, dil bilgisel ve kelime zenginliğine dayalı özellik vektörleri kullanılarak Türkçe dokümanlar için ilk defa işlevsel kelimelerin frekansları çıkarılmıştır. 18 yazara ait 35 farklı doküman kullanılarak NB, DVM, RO, K-En Yakın Komşu (KEK), Çok Katmanlı Algılayıcı (ÇKA, Multiple Layer Perceptron) ve Öz Düzenleyici Özellik Haritası (ÖÖH) yöntemleri kullanılmış olup %89,2 ile ÇKA yönteminde en başarılı sonuca ulaşılmıştır [20].

Örücü, 2009 yılındaki çalışmasıyla 234.067 eser kullanarak büyük ölçekli bir Türkçe külliyat oluşturarak Türk diline ait karakteristikleri keşfeden bir uygulama geliştirmiştir. Daha sonra bu uygulamayı kullanarak birçok kelime ve harf bazlı analiz ile 16 yazara ait toplam 33.666 eser için yazar tanıma işlemi gerçekleştirilmiş ve %87'lik başarıya ulaşılmıştır. Bu çalışma kullandığı eser sayısı bakımından en büyük çaplı Türkçe metin sınıflayıcısı olmuştur [21].

Varol 2011 yılındaki çalışmasıyla Türkçe dokümanlardaki ilk şair tanıma çalışmasını gerçekleştirmiştir. Şair tanımadaki zorluklardan ve yazar tanıma ile arasındaki farklılaşmalardan bahsetmiştir. Ayrıca eski dille yazılmış şiirlerin tamamen ayrı bir çalışmayı gerektireceğinden ötürü çalışmasında Osmanlı Türkçesi şiirler yerine günümüz Türkçesiyle yazılmış şiirlere yer vermiştir. Çalışmada her bir şairden 30 şiir olmak üzere toplam 210 eser incelenmiştir. Kelimelerin dilbilgisi özellikleri, karakter n-gramları gibi bazı özellik vektörleri ve "Ng-ind" isminde bir sınıflandırma yöntemi ile %71'lik başarı oranına ulaşılmıştır [5].

Yasdi ve Diri, çalışmalarında (2012) kelime kökleri ve 2'li n-gram ile özellik vektörü oluşturup Soyut Özellik Çıkarımı (SÖÇ) yöntemi ile analiz etmişlerdir. Elde edilen sonuçları Temel Bileşen Analizi (TBA), Korelasyon Tabanlı Özellik Seçici (KTÖS, Correlation Based Feature Selection) ve ki-kare gibi yöntemler ile karşılaştırarak SÖÇ yönteminin başarısını test etmişlerdir. Sonuç olarak SÖÇ yöntemi ile İngilizce ve Türkçe veri setleri için başarılı sonuçlar elde edilmiş ve bu yöntemin dilden bağımsız bir şekilde başarılı sonuçlar verdiğini göstermişlerdir [22]. Daha sonra Kolyiğit ve arkadaşları 5 yazara ait toplam 350 eserden

oluşan bir derlem hazırlayarak KEK yöntemini kullanıp Türkçe dokümanlar için yazar tanıma sistemi geliştirmiş ve %77 oranında başarı elde etmişlerdir [23].

Levent ve Diri (2014)'de yapay sinir ağları (YSA) yöntemi ile yazar tanıma için bir sistem geliştirmişlerdir. Sistemi üç farklı veri seti ile test etmişlerdir. İlk olarak 16 farklı yazarın 50'şer gazete makalesi, ikinci olarak 10 kadın ve 10 erkek yazardan oluşan toplam 400 makale, son olarak aynı kategorideki 16 yazarın her birine ait 40 doküman veri seti olarak belirlenmiştir. İlk veri setinde %73, ikinci veri setinde %75, son veri setinde ise %83'lük başarıya ulaşılmıştır [24].

Yazar tanıma konusundaki önemli çalışmalardan biri de 2017 yılında Kuzu tarafından gerçekleştirilmiştir. Kuzu çalışmasında çevrimiçi sosyal platformlarda Latent Semantik Analiz (LSA), TBA, Bağımsız Bileşen Analizi (BBA) yöntemlerini kullanarak yazar tanıma işlemi gerçekleştirmiştir. Bu çalışma LSA yönteminin ilk defa Türkçe yazar tanıma alanında kullanılmasından ötürü literatür içerisinde önem teşkil etmektedir. Türkçe dışında İngilizce ve Portekizce verilerle de test edilerek kullanılan yöntemlerin dilden bağımsızlığı test edilmiştir. Sonuç olarak iki farklı sosyal platform kullanılmış olup ilkinde %98.5 ikincisinde %87.9'luk bir başarı elde edilmiştir. Portekizce verilerde %81.2, İngilizce verilerde ise %83.3'lük bir başarı oranına ulaşılmıştır [25].

1.5. Ön İşlemler

1.5.1. Dokümanların Toplanması ve Standartlaştırılması

Dokümanların toplanması ve standartlaştırılması kurulacak modelin başarısını etkileyen ilk ve en önemli aşamadır. Çünkü veri seti ne kadar kaliteli ve düzgün olursa, analiz sonuçları da o kadar kaliteli ve başarılı olur. Veri setinin eksik, yetersiz, düzeltilmemiş olduğu durumlarda hangi analiz yöntemi uygulanırsa uygulansın modelin başarısı düşük ve yanlı olacaktır [26].

Araştırılan konuya ait dokümanlar pdf, txt, doc, xml, html gibi farklı dosya uzantılarına sahip olabilmektedir. İlk olarak bu dokümanların hepsinin tek ve ortak bir formata çevrilmesi gerekmektedir. Bu işlem dosya sayısı az ise manuel olarak yapılabilirken dosya sayısının fazla olduğu durumlarda çeşitli otomasyonlar aracılığıyla gerçekleştirilir.

Dokümanlar ortak bir formata dönüştürüldükten sonra gereksiz karakterlerin çıkartılması gerekmektedir. Özellikle web formatından derlenen verilerdeki Genişletilebilir İşaretleme Dili (XML) ve Hiper Metin İşaret Dili (HTML) etiketleri arındırılır. Verilerin, metin madenciliği yöntemleri ile analiz edilebilmesi için veri setindeki rakamlar, noktalama işaretleri ve yabancı dilden gelen karakterler çıkarılarak verilerin sadece harflerden oluşması sağlanır. Ancak bazı istisnai durumlarda rakam, nokta, virgül gibi karakterler veri seti için önem teşkil ettiğinden tutulabilir.

Bir sonraki adımda büyük, küçük harf farklılığından kaynaklanabilecek problemlerden ötürü bütün harfler büyültme ya da küçültme işlemine tabi tutularak ortak bir standarda kavuşturulur.

Standartlaştırma işleminin son adımında dokümanlar parçalarına ayrılmaktadır. Birçok çalışmada amaç, kelimeleri bulmak ve kelimeler üzerinden analiz gerçekleştirmek olduğu için dokümanlar kelimelerine kadar parçalarına ayrıştırılır [3]. Ancak bazı durumlarda hecelerin ya da harflerin birbirleri ile kullanım sıklıklarına bakmak için karakter analizleri gerçekleştirilmektedir. Bu durumda dokümanlar hecelerine ya da gerek görüldüğü takdirde karakterlerine kadar ayrıştırılır.

1.5.2. Yazım Hatalarının Tespiti ve Düzeltilmesi

Yazım hatalarının tespiti (YHT) ve düzeltilmesi oluşturulacak modelin yanlış öğrenimini azaltıp başarı oranını yükselten önemli bir aşamadır. YHT uzman kişiler tarafından tek tek incelenerek yapılabildiği gibi bazı istatistiksel yöntemlerle de yapılabilir. Bu yöntemler genel olarak skor değerli ve sözlük tabanlı olmak üzere iki gruba ayrılmaktadır.

Skor değerli yöntemler kelimelerin birbirleri ile çeşitli algoritmalar aracılığı ile karşılaştırılarak puan verilmesine dayanan yöntemlerdir. Bu yöntemlerden ilki ve en fazla kullanılanı 1965 yılında Levenshtein tarafından bulunan algoritmadır [27]. Kelimeler uzun olduğunda başarılı sonuçlar verirken kelime uzunlukları azaldığında yöntemin başarı oranı düşmektedir.

Sözlük tabanlı yöntemlerde ise kelimelerin her biri sözlükteki bütün kelimeler ile karşılaştırılır ve sözlükteki hiçbir kelime ile eşleşmemiş ise yazım hatalı olarak kabul edilir ve skor değerli yöntemler ya da uzman kişiler tarafından doğru kelime ile düzeltilir. Literatürde en sık kullanılan sözlük tabanlı yöntemler Hunspell, Aspell, Ispell ve Myspell

yöntemleridir [28]. Bu yöntemlerde sözlükle karşılaştırma gerçekleştirildiğinden veri setinde kullanılan dilin yapısı önem teşkil etmektedir. İlk Türkçe sözlük tabanlı yöntem Akın tarafından Hunspell'in Türkçeleştirilmesi ile "tr-spell" isminde gerçekleştirilmiştir. Ancak Hunspell Macarca kökenli olduğu için çeviri aracılığıyla yapılan bu uygulama çok başarılı sonuçlar vermemiştir. Bu sebepten daha sonra Akın ve arkadaşları tarafından Türkçede kullanılmak üzere günümüzde de kullanılan Zemberek projesi geliştirilmiş ve daha başarılı sonuçlara ulaşılmıştır [29].

1.5.3. Etkisiz Kelimelerin Çıkarılması

Etkisiz kelimelerin çıkarılması (Stop word removal), veri seti içerisinde sıklıkla geçen fakat doküman için ayırt edici özelliği olmayan kelimelerin veri setinden çıkarılması işlemidir. Bu işlem için etkisiz kelime listesi oluşturulur ve bu listede geçen kelimeler veri setinden çıkarılır [30].

Etkisiz kelimelerinin çıkarılmasının gerekliliği bu konudaki çalışmaların ilk günlerinden beri bilinmektedir. Bu kelimelerin sisteme eklenmesiyle, geliştirilecek yöntemlerin başarılarının artacağı öngörülmektedir. Ayrıca etkisiz kelimelerin sistemden çıkarılmaması sistemin eğitim aşamasının yavaşlamasına neden olmaktadır [30].

Etkisiz kelime listesi genellikle her uygulama için özel hazırlanır. Bazı istisnai durumlarda daha da ayrıntıya girilip her bir doküman için ayrı bir şekilde hazırlandığı da olur. Ancak genellikle tek bir etkisiz kelime listesi oluşturulup bütün dokümanlar için kullanılır. Etkisiz kelimeler, istatistiksel yöntemlerle bulunabilmesine karşın genellikle uzman kişiler tarafından hazırlanmaktadır.

Bazı çalışmalarda kelime uzunluğu önceden belirlenen bir değerden küçük olan bütün kelimeler de etkisiz kelime olarak düşünülebilir. Genellikle iki ya da üç karakter olarak belirlenen bu değer özellikle "ve, de, da, ki, ile, ya da" gibi bağlaçları modelden çıkarmak için tercih edilir [26].

1.5.4. Gövdeleme

Gövdeleme (stemming), kelimelerin anlamlı en alt birimi olan köklerine kadar indirgenmesine verilen isimdir. Bu sebepten literatürde kök bulma şeklinde de ifade

edilmektedir. Bu alandaki ilk çalışmayı 1968 yılında İngilizce için Lovins gerçekleştirmiştir [31]. Lovins'in çalışması, bilim dünyasında oldukça ilgi uyandırıp yeni bir alan olan metin madenciliği alanının açılmasına öncülük etmiştir. Lovins'in çalışmasından sonra 1980 yılında Porter kendi adını taşıyan algoritmasını yayınlayarak bu alandaki en başarılı çalışmalardan birisini gerçekleştirmekle geniş kitlelere ulaşmıştır [32]. Algoritma bazı değişiklikleri ile beraber günümüzde bile en çok kullanılan gövdeleme yöntemlerinden biri olmuştur.

Gövdeleme yöntemlerinin en zayıf yönü her dilin farklı kuralları olduğu için dile özel farklılıklar göstermesidir. Türkçe sondan eklemeli bir dil olduğu için gövdeleme işlemi açısından oldukça zordur. Kelimeye gelen eklerden yapım ekleri kelimenin anlamını değiştirdiğinden gövdeleme sırasında atılmamalı, çekim ekleri ise atılmalıdır. Türkçenin yapısı gereği aynı ses değerine sahip ekin hem çekim eki hem de yapım eki olabilmesi gibi nedenlerden ötürü başarılı bir gövdeleme algoritması yapmak oldukça zordur. İsim çekiminde kullanılan "-de" hal ekinin kalıplaşma yolu ile "gözde" kelimesini türetmesi örnek olarak verilebilir.

İngilizce ve Türkçedeki dil yapısının zorluklarını göstermek için Tablo 1'de "yapmak" kelimesinin Türkçe ve İngilizce karşılaştırılması gösterilmiştir [33].

Tablo 1. Türkçe ve İngilizce Kelime Yapısı Karşılaştırması

Türkçe	İngilizce
Yap	do
Yapmak	to do
Yapıyorum	I'm doing
Yapacağım	I'll do
Yapabilirim	I can do
Yapardım	I used to do
Yapacaktım	I was going to do
Yapalım	let's do
Yapınca	when I do
Yapmadan	without doing
Yaptığım	the one which I did
Yaparak	by doing

Tablo 1'den anlaşılacağı üzere İngilizcede kelimelere farklı anlamlar yüklemek için yeni kelimeler kullanılırken Türkçede mevcut kelimeye ekler getirilmektedir. Bu nedenle Türkçe gibi sondan eklemeli dillerde kelimenin kökünü bulmak oldukça zor bir işlemdir. Bu bilgiler ışığında Türkçe çalışmalar için kullanılabilen gövdeleme yöntemleri aşağıda yer almaktadır.

1.5.4.1. Sıfır Gövdeleme

Bu yöntemde kelimeler hiçbir gövdeleme işlemine tabi tutulmadan olduğu gibi sisteme dâhil edilir. Özellikle yabancı kelimelerin sıklıkla geçtiği çalışmalarda gövdeleme işlemleri neticesinde kelimenin köküne indirgeme işlemi doğru bir şekilde gerçekleştirilemeyebilir. Bu nedenle gövdelemenin hiç yapılmaması daha başarılı sonuçlar verebilmektedir [2].

Literatürde ilk metin madenciliği çalışmalarında gövdeleme uygulanmamışken ilerleyen zamanda başarılı gövdeleme yöntemlerinin bulunması ile beraber bu yöntemin geçerliliği azalmıştır. İstisnai durumlar dışında kelimenin herhangi bir ek aldığında farklı bir kök olarak algılanmasından ötürü diğer yöntemlere göre başarı oranı düşüktür.

1.5.4.2. Sabit Önekli Gövdeleme

Bu yöntemde kelimenin önceden belirlenen harf sayısı kadar karakteri gövde olarak belirlenir ve kelimenin geri kalan kısmı atılır. Örnek olarak harf sayısının 5 olarak belirlendiği bir durumda "deneme" ve "metnidir" sözcüklerinin kökleri sırasıyla "denem" ve "metni" olacaktır.

Sabit önekli gövdeleme yaklaşımı, özellikle Türkçe gibi sondan eklemeli dillerde gerçekleştirilen çalışmalarda dilin gramer özelliklerine girmeden kelimenin gövdesinden ekini çıkarmayı amaçladığı için tercih edilmektedir. Türkçe için ilk olarak 1981 yılında Köksal tarafından geliştirilmiştir. Ayrıca Köksal'ın geliştirmiş olduğu yöntem Türkçe gövdeleme konusunda yapılmış ilk çalışmadır. Bu çalışmada kelimenin ilk 5 harfi gövde olarak belirlenmiştir. Bu karara birçok kelimenin incelenmesi neticesinde varılmıştır [34].

2002 yılında Altıntaş ve Can tarafından yapılan bir çalışmada Milliyet Gazetesi'nde 1 Ocak 1997 ile 12 Eylül 1998 yılları arasında yayınlanan yaklaşık 18 milyon kelimelik metinler incelenmiştir. İnceleme neticesinde Türkçe metinler için gövde uzunluğu

ortalaması tüm kelimeler için 4,58 olarak bulunmuştur. Yinelenen kelimeler çıkartıldıktan sonra özgün kelimeler için ortalama gövde uzunluğu 6,58 olarak belirlenmiştir [35]. Benzer nitelikte bir çalışmayı yine aynı yıl Dalkılıç ve Çebi gerçekleştirmiştir. Yaklaşık yarısını Türkiye Parlamentosundaki konuşmaların, geri kalan kısmını da çeşitli dergi, magazin ve hikâyelerin oluşturduğu yaklaşık 50 milyon kelime içeren veri setinde ortalama kelime uzunluğu 6,20 karakter olarak tespit edilmiştir [36]. Bu bilgiler ışında Sever ve Tonta, Türkçe için yapmış oldukları çalışmada 5, 6 ve 7 karakterli sabit gövdelemelerin Türkçe için hızlı ve uygun olduğunu belirlemiştir [37].

Sabit önekli gövdeleme karakter analizi için n-gram yöntemine benzemesine karşın farklı bir yaklaşımdır. Sabit önekli gövdelemede kelimenin belirtilen karakterden sonraki kısmı atılarak gövde bulunurken, n-gram yönteminde kelimenin n sayıda alt kökleri gövde olarak belirlenmektedir [38]. Örneğin "Trabzon" kelimesinin 4 karakterli sabit gövdesi sadece "trab" şeklindedir. Ancak 4-gramları ise "trab", "rabz", "abzo", "bzon" şeklindedir.

1.5.4.3. Ek Çıkarımlı Gövdeleme

Bu yöntemde kelimeler kullanılan dilin gramer kuralları dikkate alınarak geliştirilen algoritmalar tarafından eklerinden ayrıştırılarak köklerine indirgenir. Bu işlem ön ekleri fazla olan diller için soldan sağa, son ekleri fazla olan diller için ise sağdan sola gerçekleşmektedir. Türkçe sondan eklemeli bir dil olduğu için geliştirilen gövdeleme yöntemleri sağdan sola ek çıkarma şeklindedir.

Türkçe için ek çıkarımlı gövdeleme konusundaki en yaygın yöntem 1999 yılında bu alandaki önde gelen bilim insanlarından Oflazer tarafından geliştirilmeye başlanmıştır [39]. Eryiğit ve Adalı onun çalışmalarından esinlenerek 2004 yılında Türkçe Morfolojik Analizini gerçekleştirmişlerdir [40]. Çilden, bütün bu çalışmalardan esinlenerek 2006 yılında Snowball Algoritmasının Türkçe kısmını hazırlayarak algoritmayı tamamlamıştır. Snowball Algoritması, Türkçe gibi 15 farklı dile hizmet veren Porter algoritmasından esinlenerek hazırlanmış ek çıkarımlı gövdeleme yöntemidir. Bu alanda farklı yöntemlerde geliştirilmesine rağmen literatürde yaygın olarak Snowball Algoritması kullanılmaktadır.

Ek çıkarımlı gövdeleme yöntemlerinin avantajı herhangi bir sözlüğe bakmadan çalışmasıdır. Bir sözlük kontrolü yapılmadığından oldukça hızlı çalışmaktadır. Dezavantajı ise yine sözlüğe bakılmamasıdır. Sözlük kontrolü olmadığından kelimelerin dilinin farklı olup olmadığı, yazım yanlışının bulunup bulunmadığı kontrol edilememektedir.

1.5.4.4. Sözlük Tabanlı Gövdeleme

Bu yöntemde ise kelimelerin gövdeleri sözlükte bulunan bütün kelimeler ile karşılaştırılarak gerçekleştirilir. Her kelime için sözlükteki bütün kelimeler ile karşılaştırma yapıldığından oldukça yavaş ama etkin bir yöntemdir.

Türkçe için ilk sözlük tabanlı gövdeleme yöntemi 1995 yılında Alpkoçak ve arkadaşları tarafından ortaya atılan en uzun eşleşme yöntemidir. Bu yöntemde kelime önce bir sözlükte aranır, bulunamadığı durumda sonundan bir harf silinerek arama işlemi yeniden yapılır. Süreç bir gövdenin bulunması ya da kelimenin bir harf kalması durumunda sona erer [41]. Kullanım olarak çok basittir ancak kelime ile ilgisiz gövdelerin bulunma ihtimali de vardır. Ayrıca kökün ünsüz düşmesi ve yumuşaması gibi durumlarında başarısız sonuçlar vermektedir.

İlerleyen zamanda sözlük tabanlı birçok algoritma ortaya çıkmıştır. TÜBİTAK tarafından 2007 yılında geliştirilmeye başlanan Zemberek Kütüphanesinin Gövdeleme Algoritması bunlar içerisinde en başarılı sonuç veren yaklaşımlardan birisi olmuştur [29]. YHT için de kullanılan Zemberek Algoritması, günümüzde özgür yazılımcılar tarafından devam ettirilen sözlük tabanlı bir gövdeleme yöntemine sahiptir.

1.5.5. Kelime Grupları

Metin madenciliği sürecinde bir terim, genellikle bir kelimeden oluşur. Fakat bazı durumlarda kelimelerin birlikte kullanımları da önemli olabilir. Bu nedenle terimler oluşturulmadan kelime grupları incelenip önemli olduğu düşünülenlerin tespit edilip terim olarak eklenmesi sistemin başarı oranını arttıracaktır.

Kelime grupları uzman kişiler tarafından belirlenebileceği gibi n-gram adı verilerin istatistiki bir yöntem ile de tespit edilebilir. Bu yöntemde ilk olarak her kelime kendinden sonraki kelime ile birleştirilip ikili kelime grubu (bigram) oluşturulur. Oluşturulan bigramların terim olarak eklenip eklenmeyeceğine iki şekilde karar verilir. İlk yöntemde mevcut bir sözlükle karşılaştırılarak bigramın terim olarak kullanılıp kullanılmayacağına karar verilirken ikinci yöntemde ise bigramların sıklıklarına bakılır. İstatistiksel olarak yüksek sıklığa sahip bigramların önemli olduğu varsayılarak terim olarak ekleme işlemi gerçekleştirilir [42].

N-gram kelime grupları analizlerinde unutulmaması gereken bir konu da bigramlar oluşturulurken kelimelerin kendilerinin bulunduğu terimlerin sistemden silinmeyeceğidir. Onlara bigramlar eklenir. Dolayısıyla veri setinde bir azalma yaşanmaz, aksine bir şişme gerçekleşir. Bu nedenle n-gramların etkisi anlamlı bulunmazsa modelin şişmesine neden olacağından dâhil edilmez.

Kelime grupları istenilen durumlarda üçlü (trigram) ya da dörtlü gruplar halinde de incelenebilir. Bütün kelimelerin incelenmesi tamamlandıktan sonra terim ekleme işlemi de tamamlanmış olur [42].

1.5.6. Karakter Analizi

Metin sınıflandırma işlemleri için her zaman kelime analizleri yeterli olmamaktadır. Bazı durumlarda kelimelerin hecelerine hatta karakterlerine inmek başarı oranını arttırabilir. Hece analizi gerçekleştirilirken veri setinin sahip olduğu dilin kendi yapısı yanış sıra kelime gruplarında olduğu gibi n-gram yönteminden de yararlanılabilir.

Karakter analizi için n-gram, bir karakter dizisinin n boyutundaki karakter dilimlerine ayrılan ve kullanım sıklığına dayanan bir işlemdir. N-gram yöntemi kullanılan kelime sayısına göre 2-gram, 3-gram ve 4-gram şeklinde ifade edilmektedir [43]. Örnek olarak "Deneme metni" cümlesinin n-gram'ları;

2-gramlar: "de", "en", "ne", "em", "me", "e_", "_m", "me", "et", "tn", "ni"

3-gramlar: "den", "ene", "nem", "eme", "me_", "e_m", "_me", "met", "etn", "tni"

4-gramlar: "dene", "enem", "neme", "eme_", "me_m", "e_me", "_met", "metn", "etni"

şeklinde çıkarılır. Burada çıkarılan her bir karakter grubu terim olarak düşünülerek metin sınıflandırma işlemi gerçekleştirilir.

1.5.7. Kelime Filtreleme

Metin sınıflandırma işleminde veri setinde çok sayıda ayırt edici özelliği düşük kelime bulunur. Bu kelimeler analizlerin gerçekleştirme süresini ve veri setinin boyutunu arttırdığı gibi, başarı oranını da düşürmektedir. Dolayısıyla modeli olumsuz etkileyen bu kelimelerin sistemden çıkarılması kurulacak modeli her anlamda daha başarılı yapacaktır [44].

Filtrelenecek kelimelerin tespiti önceden belirlenen bir sıklığa sahip kelimelerin çıkarılmasıyla gerçekleştirilir. Bu sayı sistemde bulunan veri setinin boyutuna göre 3, 5, 10, 20 şeklinde olabilmektedir.

Kelime filtreleme ve etkisiz kelime yöntemleri birbirlerine benzemesine rağmen önemli bir farklılıkları bulunmaktadır. Etkisiz kelime yönteminde yüksek sıklığa sahip olmasına rağmen ayırt edici özelliği düşük kelimeler sistemden çıkarılırken kelime budama işleminde ayırt edici özelliğine bakılmaksızın sadece düşük sıklığa sahip kelimeler sistemden çıkarılmaktadır.

1.6. Ağırlıklandırma

Metin sınıflandırmanın önemli aşamalarından bir tanesi de metin formatında tutulan verilerin ağırlıklandırılarak sayısal formata dönüştürülmesidir. Bu işlem için Kelime Çantası (KÇ, Bag Of Words) ve Vektör Uzayı (VU, Vector Space) modelleri en bilindik yaklaşımlardır [45].

1.6.1. Kelime Çantası

KÇ metin madenciliği çalışmalarında kelimelerin basit bir şekilde sayısallaştırılmasını sağlayan bir yöntemdir. İlk olarak 1959 yılında Harris tarafından ortaya çıkmıştır [46]. Kelimelerin sırası ve dil bilgisi yapısından bağımsız bir şekilde yalnızca dokümanda geçme sıklığına bakılarak her bir kelimeye ağırlık değeri verilmesine dayalı bir yöntemdir. Bazı durumlarda sıklıklar yerine kelimenin bulunup bulunmamasına göre ikili (varlık/yokluk ya da 1/0) ağırlıklar da kullanılmaktadır [2]. KÇ genelde doküman sınıflama işleminde, her bir kelimenin terim olarak kullanılmasını temel alır.

Tablo 2. Örnek Dokümanlar

No	Doküman
1	bu bir denemedir bu
2	bu başka bir denemedir
3	bir iki bir iki
4	sadece

KÇ sırasız bir belge temsili olduğundan kelimelerin hangi sırada olduğunun bir önemi yoktur sadece kelimelerin hangi dokümanda ne kadar sayıda geçtiğine bakılır. Bu bilgiler ışığında Tablo 2'de belirtilen dokümanlara KÇ modeli uygulandıktan sonra kelimelerin sayısallaştırılmış durumları Tablo 3'de gösterilmiştir [26].

Tablo 3. KÇ Sonuçları

Doküman	başka	bir	bu	denemedir	iki	Sadece
bu bir denemedir bu	0.00	1.00	2.00	1.00	0.00	0.00
bu başka bir denemedir	1.00	1.00	1.00	1.00	0.00	0.00
bir iki bir iki	0.00	2.00	0.00	0.00	2.00	0.00
sadece	0.00	0.00	0.00	0.00	0.00	1.00

Tablo 3'de KÇ modeline ait bir örnek yer almaktadır. Bu yöntem basit olduğu için oldukça hızlı çalışmaktadır. Ancak kelimelerin doküman için önemli olup olmadığına bakmadığı için başarı oranı düşüktür. Başarı oranını yükseltmek için Vektör Uzayı (VU) modeli kullanılmaktadır.

1.6.2. Vektör Uzayı

VU metin verilerine özel çok boyutlu vektörler oluşturan cebirsel bir yöntemdir. Metinleri vektör olarak ifade edip bu vektörler arasındaki açıya bakarak birbirleri arasındaki uzaklıklar hesaplanmaktadır. Bu uzaklık değerleri neticesinde kelimelerin birbirlerine karşı skor değerleri hesaplanarak modelleme gerçekleştirilir [2].

Terimin metin içerisindeki değerini ifade etmenin pek çok yöntemi mevcuttur. Bu yöntemlerden en çok kullanılanı terim sıklığı (TS) ve ters doküman sıklığı (TDS) özelliklerinin beraber kullanıldığı "TF-IDF" ağırlıklandırma yöntemidir. TF-IDF kısaltması TS ve TDS ifadelerinin İngilizceleri olan "term frequency and inverse document frequency" ifadesinin baş harflerinden gelmektedir. Yöntem sırasında TS ve TDS beraber uygulandığı için bu ismi almıştır. Ayrıca daha doğru sonuçlar elde edebilmek için normalleştirme işlemi de uygulanmaktadır.

1.6.2.1. Terim Sıklığı

Terim sıklığı (TS), her bir terimin ilgili dokümanda bulunma sayısına bağlı bir ağırlıklandırma işlemi olarak tanımlanmaktadır. İlgili kelimenin yalnızca bulunup bulunmaması ile oluşturulabildiği gibi logaritmik fonksiyonlar kullanılarak karışık bir şekilde de oluşturulabilir. Bu tamamen dokümanların içeriğine, uygulanacak veri madenciliği yöntemine ve istenilen amaca göre değişiklik gösterir. Literatürde sık kullanılan TS yöntemleri Tablo 4'de gösterilmiştir [2].

Tablo 4. TS Yöntemleri

Yöntem	Formül
Doğal	$tf_{t,d}$
Sıklık	$tf_{t,d} / N$
Logaritmik	$1 + \log_{10}(tf_{t,d})$
Arttırılmış	$0{,}5 + \frac{0{,}5 \times tf_{t,d}}{\max_t(tf_{t,d})}$
Boole	$\begin{cases} \text{eğer } tf_{t,d}>0 & \rightarrow 1 \\ \text{diğer durumlarda} & \rightarrow 0 \end{cases}$
Logaritmik Ortalama	$\frac{1+\log_{10}(tf_{t,d})}{1+\log_{10}(\frac{\sum tf_{t.d}}{N})}$

1.6.2.2. Ters Doküman Sıklığı

Modelleme sırasında yalnızca TS yöntemini kullanmak her terimin diğer dokümanlarda bulunma sıklığından bağımsız bir şekilde aynı ağırlığı almasına neden olur. Başka bir ifade ile bir terimin hangi dokümanı daha çok temsil ettiğini gösterebilmek için, veri kümesindeki tüm dokümanlardaki sıklık değerlerine bakmak gerekmektedir. Örneğin "araba endüstrisi" konulu dokümanların sınıflandırıldığını varsayalım. Bu durumda bütün dokümanlarda "araba" kelimesinin sıklığı yüksek ancak ayırt ediciliği az olacaktır. Literatürde en sık kullanılan Ters Doküman Sıklığı (TDS) yöntemleri Tablo 5'de gösterilmiştir [2].

Tablo 5. TDS Yöntemleri

Yöntem	Formül
Yok	1
TDS	$\log_{10} \frac{N}{df_t}$
Olasılıksal TDS	$\max\left\{0, \log_{10} \frac{N-df_t}{df_t}\right\}$

1.6.2.3. Normalleştirme

Büyük dokümanlarda bulunan kelimeler küçük dokümanlara göre daha çok sıklığa sahip olmaktadır. Bu sebepten özellikle küçük ve büyük dokümanların beraber olduğu durumda normalleştirme yapılmazsa VU oluşumu olumsuz etkilenir. Büyük dokümanlardaki küçük sıklığa sahip kelimeler küçük dokümandaki büyük sıklığa sahip kelimelerden daha önemli gözükür. Bu durumu ortadan kaldırmak için normalleştirme işlemi yapılır. Literatürde en sık kullanılan normalleştirme yöntemleri Tablo 6'da gösterilmiştir [2].

Tablo 6. Normalleştirme Yöntemleri

Yöntem	Formül
Yok	1
Kosinüs	$\frac{1}{\|W_i\|}$
Eşsiz Eksen	1 / u
Boyut	1 / karaktersayısı$^{\alpha}$, $\alpha < 1$

1.7. Doğrulama

Metin madenciliğinde yazar tanıma problemi için kullanılan sınıflandırma yöntemlerinde veri kümesi test ve eğitim olmak üzere ikiye ayrılmaktadır. Eğitim veri seti ile öğrenilen veriler test veri seti ile test edilerek modelin başarısı ölçülür [47]. Bu işleme doğrulama denilmektedir. Literatürde çok sayıda doğrulama yöntemi bulunmasına rağmen en fazla kullanılan doğrulama yöntemleri ayırma ve çapraz doğrulama yöntemleridir.

1.7.1. Ayırma

Bu yöntemde veriler önceden belirlenen oranda bölünerek eğitim ve test şeklinde ayrıştırılır. Bölme işlemi için basit rastgele ya da sistematik örnekleme yöntemleri kullanılır. Ayırma yöntemi oldukça hızlı çalışır, test verileri eğitim verilerinde kullanılmadığından sonuçlar yansızdır. Ancak test verileri sadece bir defa çalıştırıldığından modelde ezberleme sorunu ile karşılaşılabilmektedir [47].

1.7.1. Çapraz Doğrulama

Birden çok doğrulamanın tekrarlandığı ve doğrulama sayısına göre ismi değişen bir yöntemdir. Toplam 5 adet doğrulama yapılacaksa “5-Doğrulama”, 10 doğrulama yapılacaksa “10-Doğrulama” şeklinde ifade edilir.

Tablo 7. 10-Doğrulama Yöntemi

Veri Sayısı (%)	Veri Setinin Ayrımı									
10%	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test
10%	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim
10%	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim
10%	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim
10%	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim
10%	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
10%	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
10%	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
10%	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
10%	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim

Tablo 7'de gösterildiği üzere 10-Doğrulama için veri setinin eğitim ve test olarak ayrıştırılması esnasında veri seti 10 eşit büyüklükte parçaya ayrılıp her bir parça ayrı bir şekilde test edilir. Bu sayede modelin verileri ezberlemesinin önüne geçilir ancak doğrulama işlemi 10 kere tekrarlandığından süreç ayırma yöntemine göre uzun sürmektedir [47].

1.8. Sınıflandırma

Sınıflandırma analizleri bir veri kümesindeki değerlerin önceden bilinen gruplar içerisinden hangisine ait olduğunu tespit etmeye yarayan yöntemlerdir [48]. Sınıflandırma analizlerine kategorilere ayrıştırılmak istenilen gruplar önceden bilindiği için gözetimli öğrenme de denilmektedir. Sınıflandırma analizleri nicel veriler ile gerçekleştirildiği için metin verilerinin sınıflandırılmasında verilerin önceden sayısallaştırılması gerekir. Bu

sebepten sınıflandırılma analizinden önce metin işleme ve doğrulama yöntemleri ile metin sayısallaştırma işlemlerinin tamamlanması gerekir.

Literatürde çok sayıda sınıflandırma analizi bulunmaktadır. Bu tez çalışmasında sınıflandırma analiz yöntemleri içerisinden KEK, NB, Karar Ağacı (KA) ve DVM yaklaşımları kullanılmıştır.

1.8.1. K En Yakın Komşu

KEK yöntemi sınıflandırılmak istenen elamanın kendinden önce sınıflandırılmış en yakın k tane elamanın sınıfına bakılarak en yakın sınıfa aktarılması şeklinde çalışan bir yöntemdir. Örneğin k=3 olduğu durumda kendinden önceki en yakın 3 elemana bakılarak en yakın olduğu sınıf değeri belirlenir. Bu yakınlık değerini hesaplamak için literatürde özelleşmiş fonksiyonlar bulunmaktadır [49]. Çalışma kapsamında bu fonksiyonlar içerisinden Kosinüs Benzerlik Ölçütü (KBÖ) ve Öklid Uzaklık Ölçütü (ÖUÖ) kullanılmıştır.

KBÖ (1)'de gösterildiği üzere hesaplanır. Formülden de anlaşılacağı üzere verilerin arasındaki uzaklığa mutlak değerce bakmaktadır [49].

$$\text{benzerlik(A,B)}: \frac{\sum_{i=1}^{n} A_i \; x \; B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \; x \; \sqrt{\sum_{i=1}^{n}(B_i)^2}} = \frac{\text{A . B}}{\|\text{A}\|.\|\text{B}\|} \tag{1}$$

ÖUÖ (2)'de gösterildiği şekilde hesaplanır. Formülden de anlaşıldığı gibi verilerin arasındaki uzaklığa karelerinin toplamının karekökü şeklinde bakmaktadır. Literatürde en fazla kullanılan uzaklık hesaplama yöntemidir [49].

$$\text{uzaklık(i,j)}: \sqrt{\left|x_{i1}-x_{j1}\right|^2+\left|x_{i2}-x_{j2}\right|^2+\ldots+\left|x_{in}-x_{jn}\right|^2} \tag{2}$$

1.8.2. Naive Bayes

NB yöntemi ismini Matematikçi Thomas Bayes'in teoreminden alan basit, hızlı ve kolay anlaşılır bir sınıflandırma algoritmasıdır. Bu yöntem, olasılık ilkelerine göre tanımlanmış bir dizi hesaplama ile sınıflandırılacak verinin en fazla olasılıkla benzerlik

gösterdiği sınıfa seçilmesine dayanır. NB yönteminin bağımsızlık önermesi (3) formüldeki şekilde hesaplanır. Burada "b" verisinin "a" sınıfına ait olma olasılığı p(a/b) şeklinde gösterilmektedir [50].

$$p(a/b) = \frac{p(b/a).p(a)}{p(b)} \quad (3)$$

NB yöntemi için en sık kullanılan dağılım (4)'de gösterildiği şekilde hesaplanan olasılık yoğunluk fonksiyonudur. Burada hangi sınıfta yer aldığı bilinmeyen "b" verisinin ait olduğu sınıfı bulmak için, her bir "a" sınıfı için (4)'de yer alan formül hesaplanır. Hesaplanan değerler neticesinde "b" verisi en yüksek olasılığa ait "a" sınıfına atanır [5].

$$p(b/a) = \prod_i p(w_i/a) \quad (4)$$

NB yöntemi için sınıflandırıcı olasılık yoğunluk fonksiyonu dışında normal, lognormal, gamma ve poisson dağılımlarıyla da oluşturulabilmektedir. Ancak en fazla kullanılan yöntem olasılık yoğunluk fonksiyonudur. Sınıflandırıcının kullanım alanı her ne kadar bileşenlerin istatistiki olarak bağımsızlık önermesi ile kısıtlı gibi gözükse de yüksek boyutlu uzay ve yeterli sayıda bu koşul esnetilerek başarılı sonuçlar elde edilir [50].

Literatürde özellikle metin verileri için NB yönteminin özelleşmiş bir durumu olan Çokterimli Naive Bayes (ÇNB, Multinominal Naive Bayes) yöntemi de kullanılmaktadır. ÇNB yöntemi NB'den farklı olarak verilerin dağılımını çokterimli bir şekilde modeller [51]. ÇNB 2003 yılında Rennie ve Shih tarafından tartışılıp, analiz edilip geliştirilerek literatüre kazandırılmıştır [52].

ÇNB modeli dokümandaki kelimeleri çok terimli modeller. Dokümanlar kelime dizisi olarak dikkate alınarak her kelimenin pozisyonunun diğerinden bağımsız oluştuğu varsayılır [52].

1.8.3. Karar Ağacı

KA, belirli bir parametreye göre verilerin sürekli olarak bölündüğü bir sınıflandırma algoritmasıdır. KA yönteminin hedefi bağımlı değişkendeki farklılıkları maksimize edecek

şekilde veriyi sıralı bir biçimde parçalarına yani farklı gruplara ayırmaktır. Bu sebepten sınıflandırma ağacı olarak da adlandırılır [53].

KA, karar düğümleri, dallar ve yapraklar olmak üzere üç bölümden oluşur. Karar düğümleri giriş verilerine sorular sorularak ağacın hangi yöne yöneleceğinin belirlendiği işlemlerdir. Dal, bu soruların cevaplarını temsil eder. Yaprak ise kategorinin bulunduğu sınıf etiketleridir. KA oluşturmak yinelemeli bir işlemdir. Ağaç, bütün verilerin oluştuğu tek bir düğüm ile başlar. Daha sonra işlemleri en iyi bölecek nitelik seçilerek ilk düğüm oluşturulur ve örnekleri bölecek nitelikler tamamlanana kadar devam edip ağaç oluşumu tamamlanır. Sonuç olarak ortaya çıkan yapraklar her bir sınıfı temsil etmiş olur [26].

Literatürde çok sayıda karar ağacı olmasına rağmen çalışma kapsamında gerçekleştirilen testler neticesinde diğer ağaç yapılarına göre daha başarılı olduğu tespit edilen C4.5 algoritması kullanılmıştır. C4.5 algoritması eksik verileri basitçe ihmal eden, sürekli verileri kullanan ve aşırı uyum sebebiyle oluşan hataları telafi eden bir ağaç modelidir. Alt ağaç değiştirme ve alt ağaç yükselme şeklinde iki ana budama stratejisi bulunmaktadır [53].

1.8.4. Destek Vektör Makinesi

Temelleri Vapnik tarafından geliştirilen DVM istatistiksel öğrenme teorisine dayalı bir eğitimli sınıflama tekniğidir. DVM'nin sahip olduğu algoritmalar başlangıçta iki sınıflı doğrusal verilerin sınıflandırma problemi için tasarlanmış olup ilerleyen süreçte çok sınıflı ve doğrusal olmayan verilerin sınıflandırılması için genelleştirilmiştir [54].

DVM'nin çalışma prensibi iki sınıfı birbirinden ayırabilen en uygun karar fonksiyonunun tahmin edilmesi şeklindedir. Başka bir ifade ile iki sınıfı birbirinden en uygun şekilde ayırabilen hiper düzlemin tanımlanması esasına dayanmaktadır [54]. DVM sınıflaması üç aşamadan oluşmaktadır. İlk aşamada eğitim hücreleri özellik vektörü olarak ifade edilir. İkinci aşamada bu özellik vektörleri kernel fonksiyonları kullanılarak özellik uzayına eşlenir. Son aşamada sınıfları en uygun şekilde ayıran n-boyutlu hiper düzlem oluşturulur [55].

1.9. Değerlendirme

Sınıflandırma analizinde kurulan modeller içerisinde hangisinin daha başarılı olduğunu öğrenmek için değerlendirme ölçeklerine bakılır. Değerlendirme ölçeklerini hesaplamak için Tablo 8'de bulunan hata matrisi tablosundan yararlanılır. Bu tez çalışması kapsamında kullanılan değerlendirme ölçekleri Doğruluk (Accuracy), Kappa Katsayısı, Duyarlılık (Recall), Hassasiyet (Precision) ve F-ölçütüdür [47].

Tablo 8. Hata Matrisi Tablosu

		Test Sonuçları	
		Pozitif	Negatif
Gerçek Durum	Pozitif	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

1.9.1. Doğruluk (Accuracy)

Sınıflandırma problemlerinde veri setinin yapısı gereği bütün verilerin önceden hangi sınıfa ait olduğu bilinmektedir. Bu sebepten test veri setine ait değerlerin gerçekte ve model sonucunda hangi sınıfa ait olduğu kolaylıkla tespit edilebilir. Bu bilgiler ışığında Doğruluk, (5)'teki gibi hesaplanarak model sonucunda tahmin edilen sınıf değerlerinin yüzde kaçının gerçekte doğru sınıfta olduğunu gösteren ölçüte verilen isimdir.

$$\text{Doğruluk} = \frac{\text{DP}+\text{DN}}{\text{DP}+\text{YP}+\text{DN}+\text{YN}} \tag{5}$$

Doğruluk en çok kullanılan değerlendirme ölçeği olmasına rağmen tek başına yeterli değildir. Özellikle veri setindeki değerlerin büyük kısmının aynı sınıfta olması durumunda yanıltıcı olabilir. Örneğin veri setinde 990'ı A, 7 tanesi B, 3 tanesi de C sınıfında olmak üzere toplam 1000 veri olsun. Bu durumda bütün veriler A grubunda tahmin edildiğinde doğruluk ölçütü %99 olacaktır. Dolayısıyla yanlış bir model kurulmasına rağmen doğruluk

ölçütü yüksek olacaktır. Bu sebepten değerlendirme gerçekleştirilirken doğruluk ölçütü ile beraber diğer ölçeklere de bakılması neticesinde daha sağlıklı sonuçlara ulaşılır [26].

1.9.2. Kappa Katsayısı

Kappa Katsayısı, (6)'da gösterildiği şekilde hesaplanmaktadır. Kappa Katsayısının yüksek değer alması model performansının yüksek olduğunu göstermektedir. Doğru yapılmış tahmin sayısının bütün tahminlere olan oranına p(a). Her bir sınıf için tahmin değerlerinin varsayıma dayalı olasılığına ise p(e) adı verilmektedir [56].

$$\kappa = \frac{\mathrm{p}(a) - \mathrm{p}(e)}{1 - \mathrm{p}(e)} \tag{6}$$

1.9.3. Duyarlılık (Recall)

Duyarlılık ölçütü (7)'deki gibi hesaplanarak yalnızca gerçek örnek arasındaki doğru sınıflandırılan örneklerin oranını vermektedir [57].

$$\text{Duyarlılık} = \frac{\mathrm{DP}}{\mathrm{DP+YN}} \tag{7}$$

1.9.4. Hassasiyet (Precision)

Hassasiyet ölçütü (8)'deki gibi hesaplanarak sınıflandırma sonuçları arasındaki doğru tahmin edilen örneklerin oranını vermektedir [57].

$$\text{Hassasiyet} = \frac{\mathrm{DP}}{\mathrm{DP+YP}} \tag{8}$$

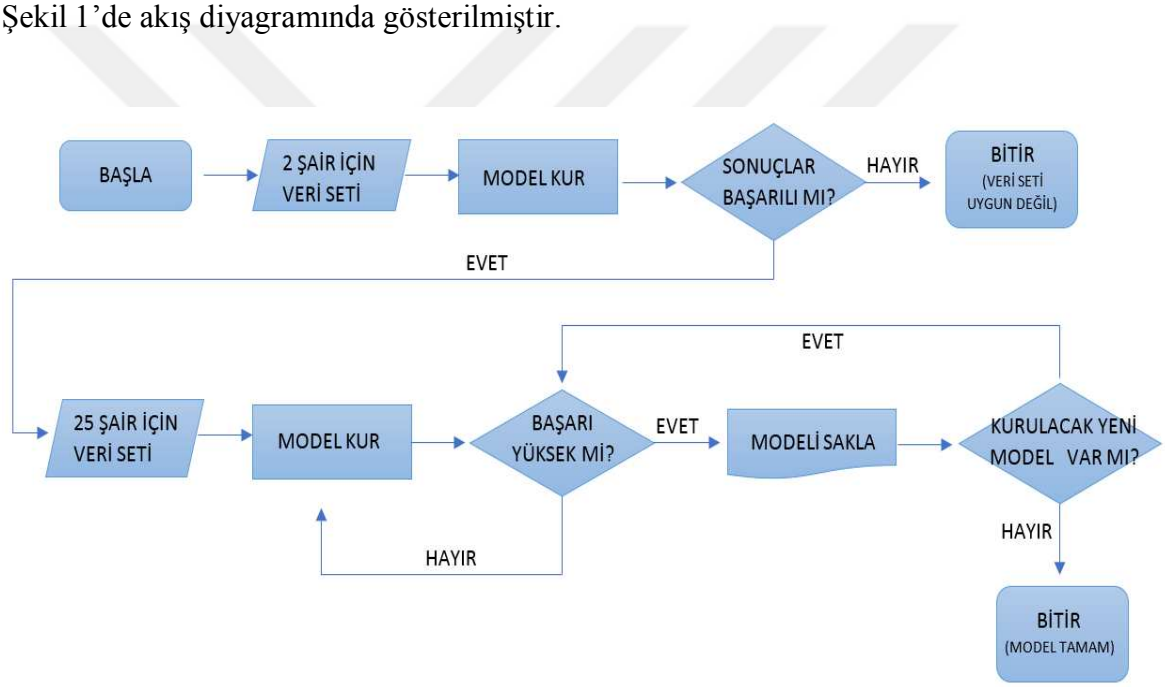
1.9.5. F-Ölçütü

Hassasiyet ve Duyarlılık ölçütlerinin harmonik ortalamasının alınması ile (9)'daki gibi hesaplanan F-ölçütü iki ölçütü de kapsadığından oldukça önemli bir değerlendirme ölçeğidir [57].

$$\text{F-ölçütü} = 2.\frac{\frac{DP}{DP+YP} \text{ x } \frac{DP}{DP+YN}}{\frac{DP}{DP+YP} + \frac{DP}{DP+YN}} \tag{9}$$

2. YAPILAN ÇALIŞMALAR

Bu tez çalışması kapsamında işleme alınan bütün divan şairlerinin eserlerini karşılaştırmadan önce metin madenciliğinin divan edebiyatı eserlerine uygun olup olmadığını görmek, uygun ise basit hatalarını ve eksiklerini tespit etmek için ilk başta iki şair seçilerek eserleri karşılaştırılmıştır. Karşılaştırma sonucunda başarılı sonuçlar elde edilmesi ile literatür kısmında bahsedilen her bir metin işleme yöntemi ve sınıflandırma analizine ait tek tek model kurulup değerlendirme ölçekleri ile başarıları sınanmıştır. Elde edilen sonuçlar neticesinde en yüksek başarıya sahip model kullanılarak divan edebiyatı için yazar tanıma işlemi gerçekleştirilmiştir. Bunun için gerçekleştirilen işlemler ana hatları ile Şekil 1'de akış diyagramında gösterilmiştir.



Şekil 1. Ana Hatları ile Yapılan Çalışmaların Akış Diyagramı

Çalışma süresince ÇNB yöntemi WEKA yazılımından yararlanılarak, Zemberek Kütüphanesi ise Netbeans 8.2 programı ile JAVA dilinde bir program yazılarak gerçekleştirilmiştir. Bu yöntemler dışındaki bütün model ve analizler Rapidminer Studio 5.3 yazılımı aracılığıyla yapılmış ve ek-1'de gösterilmiştir.

Çalışma kapsamında kullanılan divanlara Taşınabilir Belge Biçimi (PDF) formatında erişilmiştir. Ancak bu verilerin işlenebilmesi için Virgülle Ayrılmış Değerler (CSV) ya da XML gibi sistematik forma çevrilmesi gerekmektedir. Bu işlem PDF formatında tutulan verilerin standart bir şekilde tutulmamasından ötürü otomatik bir şekilde

gerçekleştirilememiştir. Bu sebepten divan verileri PDF formatından CSV formatına manuel bir şekilde kopyala-yapıştır şeklinde dönüştürülmüştür.

Veriler analiz edilirken basit analizler için 16 GB geçici belleğe (ram) sahip bilgisayarlar kullanılmıştır. İşlem sayısı fazla olan analizleri gerçekleştirmek için 32 GB geçici belleğe sahip bilgisayar kullanılmıştır. En başarılı modeli belirlemek için toplam 20 farklı model kurulmuş olup bütün analizleri tamamlamak için bilgisayarlar toplam 84 saat çalıştırılmıştır.

2.1. İki Şairin Eserlerini Karşılaştırma

İlk olarak veri setinin toplamını oluşturan 25 divan şairi içerisinden Nedim ve Şeyh Galip'in eserleri karşılaştırılmak üzere seçilmiştir. Daha sonra ilgili şairlere ait PDF formatında tutulan veriler sistematik bir format olan CSV formatına dönüştürülmüştür. Bu işlem PDF dosyasındaki şekil bozukluklarından dolayı otomatik bir şekilde yapılamamış ve oldukça zaman almıştır. Sonuç olarak Nedim'e ait 390, Şeyh Galip'e ait 448 olmak üzere toplam 838 eser içeren veri seti oluşturulmuştur.

Veri seti oluştuktan sonra ilk olarak harf karakterleri dışındaki ",-3" gibi karakterler çıkarılarak bütün harfler küçük harfe çevrilmiştir. Bu işlem sonrasında kelime uzunluğu 3'den az olan kelimeler etkisiz oldukları düşünülerek sistemden çıkarılmıştır. Kelime grupları, Karakter analizi ve Gövdeleme kullanılmamıştır. Çünkü iki şairin eserlerinin karşılaştırması işlemi bütün şairlerin eserlerinin karşılaştırması için bir ön analizdir ve hızlı bir şekilde sonuçlara ulaşmak amaçlanmıştır.

Veri setinin oluşumu ve ön işlemler tamamlandıktan sonra ağırlıklandırma için VU kullanılarak veriler sayısal formata dönüştürülmüş, doğrulama için ise 10'lu doğrulama yöntemi kullanılmıştır. Sayısal formata dönüşen veriler için gerçekleştirilen sınıflandırma analizleri sonuçları Tablo 9'da yer almaktadır.

Tablo 9. İki Şairin Eserlerinin Karşılaştırma Sonuçları

Yöntem	Özellik	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F-Ölçütü
KEK- KBÖ	k=3	95.10	0.885	94.11	94.66	94.38
KEK- ÖUÖ	k=3	96.02	0.907	95.23	95.57	95.40
NB		99.54	0.989	99.53	99.43	99.48
ÇNB		99.85	0.996	99.76	99.89	99.82
KA		94.18	0.868	93.95	93.08	93.51
DVM	c=0	98.31	0.959	97.35	98.84	98.09

Toplam 6 sınıflandırma yöntemi kullanılmış olup en yüksek başarıya ÇNB yönteminde, en düşük başarıya KA yönteminde ulaşılmıştır. Ancak hangi sınıflandırma yöntemi kullanılırsa kullanılsın iki şairin sınıflandırması için başarı oranı oldukça yüksek olmuştur.

2.2. Varsayılan Modelin Meta Verileri

İki şairin eserlerinin karşılaştırmasının başarılı sonuç vermesinin ardından divan şiirlerinde metin sınıflandırma işleminin gerçekleştirilebileceği sonucuna varılmıştır. Bu sebepten 25 divan şairine ait toplam 7148 eserden oluşan veri seti için çoklu karşılaştırma gerçekleştirilmiştir. Bu karşılaştırmada sınıflandırma verileri iki şairdeki gibi iki terimli değil çok terimlidir. Sonuç olarak 7148 esere ait şair tanımlama sistemi oluşturularak kurulan modelin başarısı test edilmiştir.

Beklenildiği üzere buradaki amaç başarı oranını yüksek tutmaktır. Bunun için her parametre değişikliğiyle beraber tekrar model kurularak test edilmelidir. Ancak bütün olası durumların test edilmesi on binlerce model kurulumuna yol açmaktadır. Ayrıca bir modelin sonuçlanmasının yaklaşık 2 saat sürdüğü göz önüne alındığında bu işlemleri gerçekleştirmek neredeyse imkânsızdır. Bu sebepten modelin süresini uzun tutmayacak varsayılan bir model kurularak bütün parametreler bu varsayılan model üzerinde değişiklik yapılarak test

edilmiştir. Söz konusu varsayılan modele ilişkin bilgileri içeren meta veriler Tablo 10'da gösterilmiştir.

Tablo 10. Varsayılan Modelin Meta Verileri

Etkisiz Kelimelerin Çıkarılması	Gövdeleme	Kelime Grupları	Karakter Analizi	Kelime Filtreleme	Ağırlıklandırma	Sınıflandırma
Kelime Uzunluğu 3'den Kısa	Sıfır Gövdeleme	Yalın Kelime	Yok	10	Vektör Uzayı	DM

2.3. Ön İşlemlerin İncelenmesi

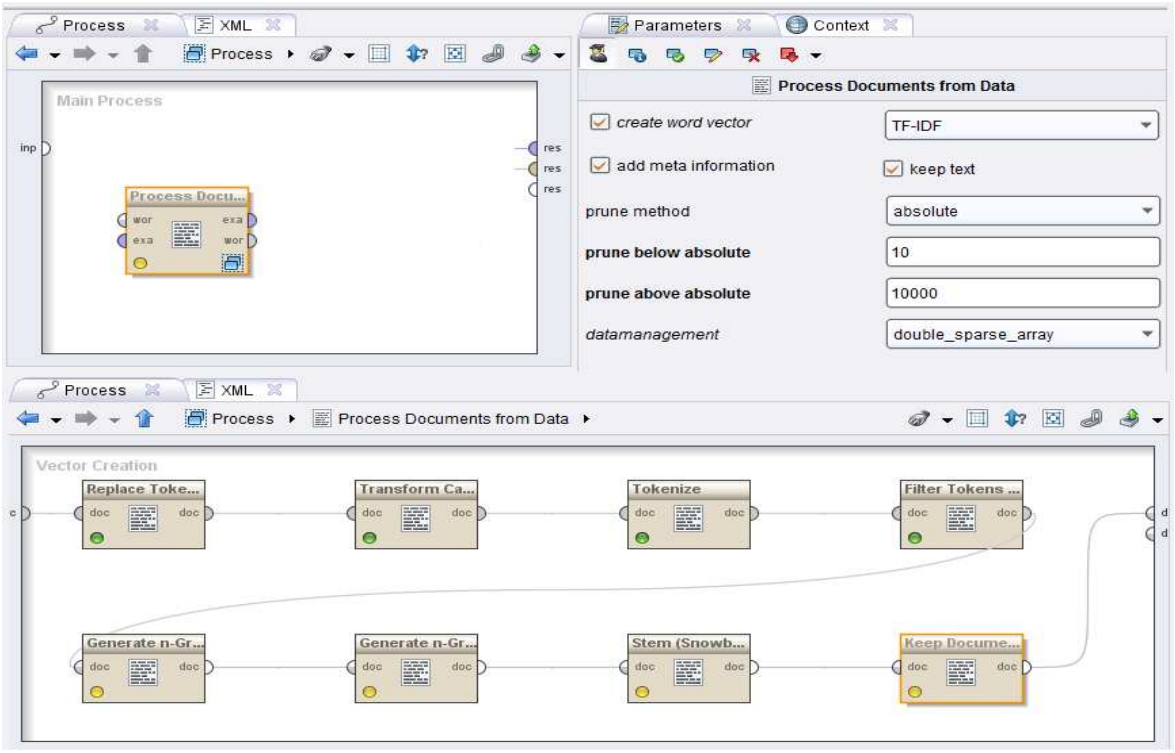
Veri setine Şekil 2'de gösterilen ön işlemler tek tek uygulanarak model performansları karşılaştırılmış ve sonuçları ilgili alt bölümlerde anlatılmıştır. Karşılaştırma yapılırken karşılaştırma yapılan işlem dışındaki bütün parametreler sabit tutulmuştur.

Ön işlemler Rapidminer programının Şekil 2'de gösterilen "Process Document from Data" işlemi ve alt işlemlerinde gerçekleştirilmiştir. Bu sürecin alt işlemlerinden "Replace Token" ile Türkçe ve İngilizcenin farklılıklarından ötürü "I ve İ" karakterleri "ı, i" karakterlerine dönüştürülmüştür. Daha sonra "Transform Cases" ile bütün harfler küçüğe çevrilip "Tokonize" işlemi ile kelimelerine kadar ayrıştırılmıştır. Son olarak "Filter tokens" ile de etkisiz kelimeler çıkarılmıştır. Bu işlemlerin hepsi bütün modellerde standart bir şekilde kullanılan işlemlerdir.

Geri kalan ön işlemlerin kullanılacak olanları için teker teker model performanslarına bakılmıştır. Model performansı yüksek olan işlemler kurulacak son modelde kullanılmış olup diğerleri kullanılmamıştır. Bütün karşılaştırmalar tamamlandıktan sonra oluşan son model Bulgular bölümünde anlatılmıştır. Performansı karşılaştırılan modellerden "Generate n-grams (Terms)" ve "Generate n-grams (Characters)" işlemleri ile kelime grupları ve karakter analizleri gerçekleştirilmiştir. Gövdeleme işlemleri için ise "Stem (Snowball)" ve "Keep Documents Parts" işlemleri kullanılmıştır. "Stem (Snowball)" ile ek çıkarımlı, "Keep

Documents Parts" ile sabit önekli gövdeleme işlemleri gerçekleştirilerek alt işlemler tamamlanmıştır.

Budanacak kelime sayısının belirlenmesi ise "Process Document from Data" işleminin kendi özelliklerinden "Prune Below Absolute" ile belirlenmiştir. Bu değer için gerekli işlemler yapılarak optimum değer belirlenmiştir.



Şekil 2. Bütün Ön işlemlerin Gösterimi

2.3.1. Yazım Hatalarının Tespiti

Veri seti birden farklı dile ait veriler içerdiğinden sözlük tabanlı yöntemler için bir çalışma gerçekleştirilmemiştir. Skor tabanlı yöntemler ise birbirine yakın puana sahip kelimelerin bile farklı anlamlarının olduğu divan edebiyatı için kullanılmamıştır.

YHT için skor tabanlı ve sözlük tabanlı yöntemler anlamlı bir başarı elde edemediğinden kullanılmamış olup veri setine ait toplam 9318 kelime uzman kişiler tarafından tek tek incelenerek bir sözlük oluşturulmuştur. Sözlük kullanılan ve kullanılmayan durumlardaki model performansları Tablo 11'de gösterilmiştir.

Tablo 11. YHT için hazırlanan Sözlük Sonuçları

Yöntem	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
Yalın	91,26	0,908	87,95	91,91	89,89
Sözlük Kullanılan	89,79	0,893	85,44	90,95	88,11

Tablo 11'de gösterilen değerlendirme ölçekleri sonuçlarına göre uzman kişiler tarafından oluşturulan herhangi bir sözlük kullanılmayan sonuçlar daha başarılı olmuştur. Bu sebeple diğer analizler herhangi bir sözlük kullanılmayan yönteme göre gerçekleştirilmiştir.

2.3.2. Etkisiz Kelimelerin Çıkarılması

Veri seti Türkçe, Arapça ve Farsça olmak üzere 3 farklı dile ait kelimeler içermesinden ötürü etkisiz kelimeler için herhangi bir dile ait literatürde bulunan etkisiz kelime listesi kullanılmamıştır. Divan edebiyatına has etkisiz kelime listesi oluşturmak özel bir çalışma gerektirdiğinden bu çalışma içerisinde herhangi bir etkisiz kelime listesi kullanılmamıştır.

Ayrı bir liste oluşturulmamasına rağmen kelime uzunluğu belirlenen düzeyden kısa olan kelimeler etkisiz olarak düşünülerek sistemden çıkarılmıştır. İdeal kelime uzunluğu gerçekleştirilen ön testler neticesinde 3 olarak belirlenmiştir. Bu sebepten sadece kelime uzunluğu 3'den küçük olanlar etkisiz kelime olarak belirlenip sistemden çıkarılmıştır.

2.3.3. Gövdeleme Yöntemleri

Gövdeleme yöntemlerinden sıfır, sabit önekli, ek çıkarımlı ve sözlük tabanlı yöntemler kullanılarak performansları karşılaştırılmıştır. Sabit önekli gövdeleme için ilk 5 ve 6 harflerin saklandığı yöntemler kullanılmıştır.

Divan edebiyatı her ne kadar Arapça, Farsça ve Eski Türkçe kelimeler barındırsa da mevcut diller içerisinden doğal olarak en çok Türkçe ile yakınlık gösterdiği için ek çıkarımlı ve sözlük tabanlı yöntemlerde Türkçe gövdeleme yöntemleri kullanılmıştır. Ek çıkarımlı gövdeleme yöntemlerinden Türkçe Snowball algoritması kullanılmıştır.

Sözlük tabanlı yöntemlerden ise Zemberek Algoritması kullanılmıştır. Zemberek Kütüphanesi kullanılarak veri setindeki toplam 9318 farklı kelimenin 2834 tanesi kelime köküne kadar ayrıştırılmıştır. 1917 adet kelime Zemberek kütüphanesinin Türkçe sözlüğünde bulunmuş fakat kelimeler yalın halde olduğu için aynen saklanmıştır. 4567 tane kelime Zemberek kütüphanesine ait Türkçe sözlükte bulunamadığından kelimeler aynen korunmuştur.

Bu bilgiler ışığında gerçekleştirilen analizlere ilişkin performans değerleri Tablo 12'de yer almaktadır.

Tablo 12. Gövdeleme Sonuçları

Yöntem	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
Sıfır	91,26	0,908	87,95	91,91	89,89
İlk 5 Harfli	90,75	0,903	87,97	91,69	89,79
İlk 6 Harfli	90,92	0,905	88,10	91,86	89,94
Ek Çıkarımlı	91,24	0,908	88,62	92,44	90,49
Sözlük Tabanlı	91,26	0,908	88,18	92,23	90,16

Tablo 12'de yer alan sonuçlara göre doğruluk değeri için gövdelemenin gerçekleşmediği sıfır gövdeleme ve sözlük tabanlı yöntemler daha başarılı olurken duyarlılık, hassasiyet ve f ölçütü değerlerinde ek çıkarımlı yöntem daha başarılı olmuştur. İlk 5 harfli ve 6 harfli gövdeleme sonuçları diğer yöntemlerden daha düşük başarı oranına sahip olmasına karşın çok büyük farklılık göstermemektedir. Değerlendirme ölçeklerinin bir kısmında sıfır gövdeleme ve sözlük tabanlı, diğerlerinde ek çıkarımlı gövdeleme başarılı olduğu için kurulacak modelde gövdelemenin önemi çok düşüktür. Bu sebeple diğer analizler gövdeleme kaynaklı yanlılığı ortadan kaldırmak için sıfır gövdeleme yöntemine göre gerçekleştirilmiştir.

2.3.4. Kelime Gruplarının Analizi

Divan edebiyatında özellikle Farsça kelimelerde "-i, -ı, -u, -ü" gibi sesli harfler ile birbirine bağlanılan kelimeler bir tamlamadır. Bu sebepten kelime gruplarının analizinde ilk olarak bu tamlamaların tek kelime olarak belirlendiği veri seti ile yalın olarak belirlenen veri setinin performans sonuçları karşılaştırılmıştır. Tablo 13'de gösterildiği üzere kelimelerin yalın olarak veri setine eklendiği durumda başarı oranı tamlamalara göre daha yüksektir. Metin sınıflandırmada kelimelerin anlamları önemli olmadığından başarı oranı yüksek modeli kullanmak doğru seçim olacaktır. Bu sebepten bütün analizler kelimelerin yalın olarak kullanıldığı veri setiyle gerçekleştirilmiştir.

Tablo 13. Veri Setine Tamlamaların Dâhil Edildiği Sonuçlar

Yöntem	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
Yalın	91,26	0,908	87,95	91,91	89,89
Tamlamalı	87,58	0,869	84,56	89,39	86,91

Tamlamaların analizi tamamlandıktan sonra kelimeler için n-gram analizleri gerçekleştirilmiştir. Bunun için kelimelerin yalın olarak tutulduğu veriler 2-gram ve Ek çıkarımlı gövdeleme ile beraber 2-gram yönteminin kullanıldığı sonuçlar karşılaştırılmış ve Tablo 14'de gösterilmiştir. Gövdeleme yöntemleri içerisinde ek çıkarımlı gövdeleme başarılı sonuç verdiği için 2-gram yöntemi ile beraber etkisi de sınanmıştır.

Tablo 14. Kelimeler için N-gram Sonuçları

Yöntem	Gövdeleme	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
Yalın	Sıfır	91,26	0,908	87,95	91,91	89,89
2-gram	Sıfır	91,45	0,910	88,13	92,44	90,23
2-gram	Ek Çıkarımlı	91,03	0,906	87,60	92,48	89,97

Tablo 14'de yer alan üç yöntemin değerlendirme sonuçları birbirine çok yakın olmakla beraber hassasiyet ölçütü dışında bütün ölçütlerde en başarılı sonuçlara sıfır gövdelemenin gerçekleştiği 2-gram yönteminde ulaşılmıştır. Bu nedenle en başarılı yöntem olarak bu yöntem seçilmiş ve son modelde kullanılmıştır.

2.3.5. Karakter Analizi

Divan şiirleri sözcüklerine kadar ayrıştırılıp analiz edildikten sonra harflerin beraber kullanım dizilimleri ve sıklıklarının analizi karakter için n-gram yöntemi kullanılarak gerçekleştirilmiştir. 2-gram ve 3-gram analizleri gerçekleştirilmiş olup analizlere ilişkin sonuçlar Tablo 15'de sunulmuştur. Burada 2-gram karakter analizlerinin sonuçlarının bütün ölçekler tarafından oldukça başarısız olduğu açık bir şekilde gözükmektedir. Fakat 3-gram karakter analizi her ne kadar karakter analizinin yapılmadığı duruma göre başarısız olsa da başarı oranı oldukça yüksektir. Ancak sonuç olarak başarı oranı en yüksek karakter analizinin gerçekleştirilmediği durumlarda saptandığı için kurulan modelde karakter analizi gerçekleştirilmemiştir.

Tablo 15. Karakter Analizi Sonuçları

Yöntem	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
Karakter Analizi Yok	91,26	0,908	87,95	91,91	89,89
2-gram	71,53	0,701	67,76	71,37	69,52
3-gram	88,44	0,879	85,79	90,61	88,13

2.3.6. Filtrelenen Kelime Sayısının Belirlenmesi

Düşük sıklığa sahip kelimelerin model oluşumuna etkisi az, ancak analizin gerçekleşme süresine etkisi fazladır. Bu, modeli başarı yönünden zayıf, oluşturma süresi bakımından hantal yapacaktır. Bu sebepten analize başlamadan sabit bir sayı belirlenip ondan düşük sıklığa sahip kelimelerin modelden çıkarılması sağlanır. Bu bilgiler ışığında

hiçbir kelimenin budanmadığı, 3, 5, 10 ve 20 sıklıktan az kelimelerin budandığı modeller kurularak değerlendirme sonuçları Tablo 16'da gösterilmiştir.

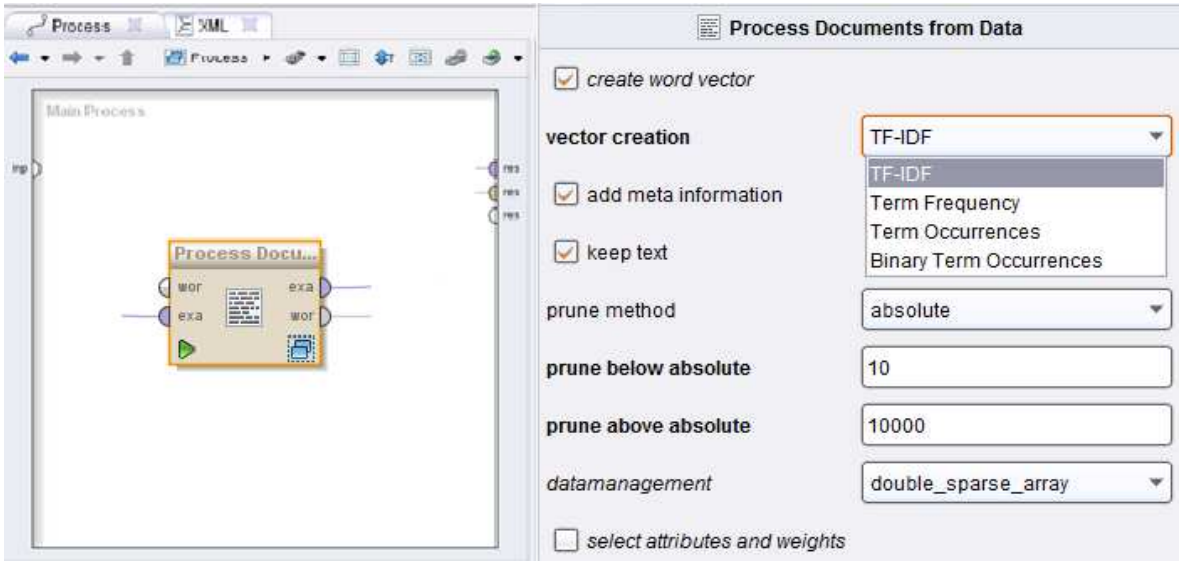
Tablo 16. Filtrelenen Kelime Sayısına Bağlı Model Performansları

Yöntem	Kelime Sayısı	Analiz Süresi	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
3'den az	26 814	06:37:03	87,35	0,867	82,14	88,10	85,02
5'den az	16 896	04:31:43	90,33	0,898	86,13	91,68	88,82
10'den az	9 318	02:53:10	91,26	0,908	87,95	91,91	89,89
20'den az	5 020	01:49:14	90,84	0,904	87,47	91,66	89,52

Tablo 16'da gösterildiği kurulan modellerin analiz süreleri, filtrelenen kelime sayısı ile ters orantılıdır. Bu sebepten en hızlı analiz 20'den az sıklığa sahip kelimelerin çıkartıldığı model ile gerçekleşmiştir. Ancak bütün performans ölçütlerine göre en yüksek başarı 10 sıklıktan az kelimelerin budandığı modelde gerçekleşmesinden ötürü kurulacak modelde bu yöntem kullanılmıştır.

2.4. Ağırlıklandırma Yönteminin Belirlenmesi

Ağırlıklandırma yöntemi Şekil 3'de gösterilen Rapidminer programının "Process Document from Data" işleminin "Vektor Creation" özelliğiyle belirlenmiştir. Burada KÇ için "term occurrences", VU için ise "TF-IDF" opsiyonları seçilerek modeller kurulmuş ve elde edilen sonuçlar Tablo 17'de gösterilmiştir.



Şekil 3. Ağırlıklandırma Yönteminin Otomasyon Üzerinde Gösterimi

KÇ yöntemi için gerçekleştirilen testler neticesinde ağırlıklandırma işlemi kelimelerin dokümanda bulunduğu sıklıklara göre belirlenimiştir. Sıklık yerine kelimenin dokümanda bulunup bulunmamasını esas alan ikili (varlık/yokluk ya da 1/0) ağırlıklar başarısız sonuç verdiğinden kullanılmamıştır.

Gerçekleştirilen denemeler neticesinde;

- Tablo 4'de belirtilen TS metotlarından sıklık yöntemi,
- Tablo 5'de belirtilen yöntemlerden TDS,
- Tablo 6'da belirtilen normalleştirme yöntemlerinden kosinüs yöntemi,

kullanılarak VU oluşturulmuştur.

Tablo 17. Ağırlıklandırma Yöntemi Sonuçları

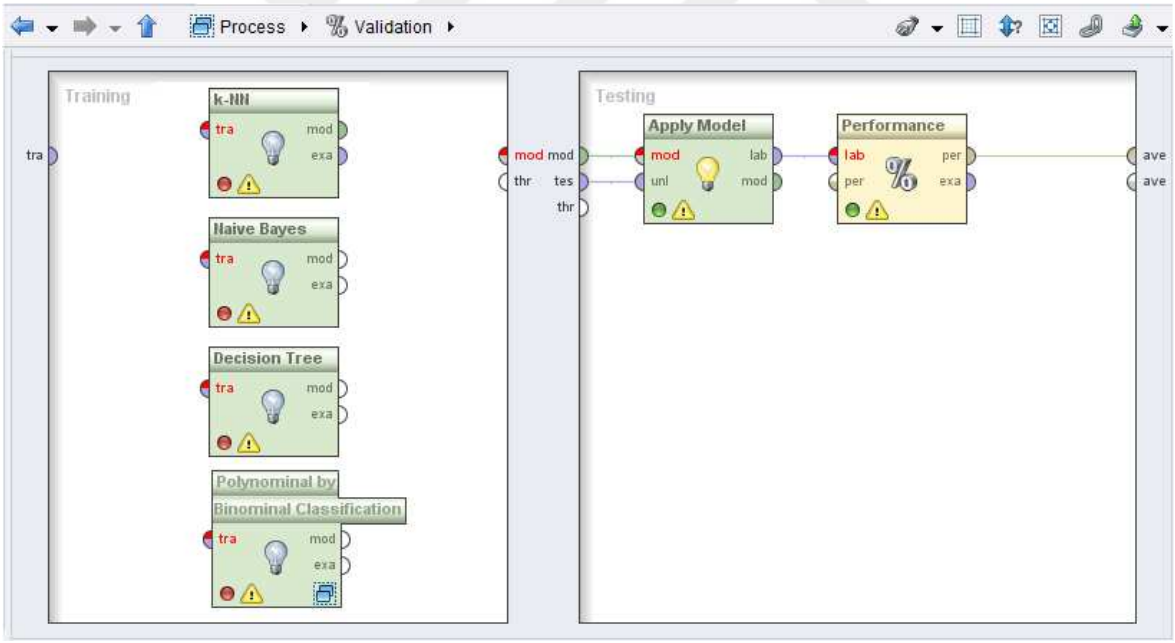
Yöntem	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
KÇ	90,28	0,898	86,06	92,76	89,28
VU	91,26	0,908	87,95	91,91	89,89

Tablo 17'de gösterilen performans ölçütü değerleri dikkate alındığında VU yönteminin doğruluk, kappa, duyarlılık ve f ölçütü için başarılı olduğu KÇ modelinin ise yalnızca hassasiyet ölçeğinde başarı oranının yüksek olduğu ortaya çıkmıştır. Her ne kadar iki yöntem arasında büyük farklar bulunmasa da hassasiyet hariç bütün performans

değerlendirme ölçeklerinde VU yöntemi daha başarılı olduğu için model kurulumu esnasında VU kullanılmıştır.

2.5. Sınıflandırma Analizinin Seçilmesi

Sınıflandırma Analizleri ceteris paribus durumda gerçekleştirilmiştir. Daha açık bir ifade ile her sınıflandırıcının model kurulumunda diğer etkileri ortadan kaldırmak için aynı ön işlemler uygulanmıştır. Sınıflandırma analizi Şekil 4'te gösterildiği üzere Rapidminer programının "% Validation" işleminin alt işlemlerinde gerçekleşmiştir. Sınıflandırmanın eğitim aşaması "Training" bölümünde, test aşaması ise "Testing" bölümünde gerçekleştirilmiştir. Eğitim aşamasında KEK algoritmaları "k-NN", NB yöntemleri "Naive Bayes", KA metotları "Decision Tree", DVM yöntemi "Polynominal by Binominal Classification" işlemleri kullanılarak gerçekleştirilmiştir. Kurulan her model yalnızca bir sınıflayıcı işlem seçilerek gerekleştirilmiştir.



Şekil 4. Sınıflandırma Analizleri Karşılaştırması

Test aşamasında "Apply Model" işlemi ile test verileri kurulan modele göre teste tabi tutulmuştur. Modelin performansı da "Performance" işlemi ile gerçekleştirilerek değerlendirme ölçütleri hesaplanmıştır. Bu bilgiler ışığında bütün sınıflandırma analizleri ayrı ayrı kurularak 7 farklı model oluşturulmuştur. Kurulan bütün sınıflandırma modellerine ilişkin analiz sonuçları Tablo 18'de gösterilmiştir.

Tablo 18. Sınıflandırma Analizi Sonuçları

Yöntem	Özellik	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
KEK- KBÖ	k=3	34,22	0,311	31,82	37,19	34,30
KEK- ÖUÖ	k=3	34,33	0,313	31,91	37,51	34,48
NB		18,40	0,167	10.48	69,99	18,23
ÇNB		75,92	0,745	61,56	65,79	63,60
KA	C4.5	79,62	0,786	77,82	79,58	78,69
DVM	c=0	91,26	0,908	87,95	91,91	89,89

Tablo 18'de gösterilen KBÖ ve ÖUÖ ölçüm yöntemleri ile hesaplanan KEK sınıflandırıcıları için bütün başarı ölçütleri %40'ın bile altında kaldığı için oldukça başarısız olmuştur. Bu yöntemlerde yapılan testler neticesinde "k" parametresinin optimum değeri "3" olarak tespit edilmiştir. Diğer "k" değerlerinde mevcut başarılara bile ulaşılamamıştır. Her ne kadar KBÖ ve ÖUÖ dışındaki diğer ölçüm yöntemleri kullanılarak KEK sınıflandırıcısı oluşturulup test edilse de başarı oranında herhangi bir artış saptanmamıştır. Bu sebepten çalışma kapsamında kullanılan veri seti için KEK sınıflandırıcısı başarısız olmuştur.

NB sınıflandırıcısına ait sonuçlar da başarısız olmuştur. Hassasiyet ölçütü dışındaki ölçüm değerleri %20'nin bile altında gerçekleşmiştir. Sadece hassasiyet ölçütü göreli olarak yüksek çıkmıştır. Fakat %69,99 olan bu değer bile hem yeteri kadar yüksek değildir hem de sadece bir ölçüm değerinin yüksek olması herhangi bir anlam taşımamaktadır. NB yönteminin özelleşmiş durumu olan ÇNB yöntemi %75,92 doğruluk ve %63,60 f-ölçütü değerleri ile kendinden önceki yöntemlere göre başarılı olurken istenilen düzeyde değildir.

KA modelleri içerisinde gerçekleştirilen ön testler neticesinde veri setine en uygun algoritmanın C4.5 olduğu tespit edilmiştir. Bu sebeple KA sınıflandırması C4.5 algoritması kullanılarak gerçekleştirilmiştir. KA sınıflandırması en yüksek değerlendirme ölçütüne %75,92 ile doğruluk değerinde ulaşmış olmasına rağmen başarı oranı diğer sınıflandırıcıların altında kalmıştır.

Tablo 18'de gösterilen sınıflandırma yöntemlerine ait başarı sonuçları incelendiğinde değerlendirme ölçütlerinin hepsine göre en yüksek başarı DVM yönteminde gerçekleşmiştir.

DVM yöntemi için gerçekleştirilen ön testler neticesinde “c” parametresinin optimum değeri “0” olarak tespit edilmiştir. Bu sebepten “c” parametresinin değeri “0” olarak alınmıştır.

İki şairin karşılaştırması sırasında bütün sınıflandırma yöntemlerinin başarı oranı yüksek gerçekleşmiş olmasına karşın, 25 şair için yapılan karşılaştırmada DVM dışındaki bütün yöntemlerin başarıları önemli düzeyde düşmüştür. Buradan yola çıkarak ilgili veri seti için şair sayısının arttığı durumda yalnızca DVM yönteminin başarılı olduğunu söyleyebiliriz. Bu sebepten kurulan modeldeki bütün analizler DVM ile gerçekleştirilmiştir.

3. BULGULAR

Divan şairlerinin eserlerine erişmek için oldukça titiz bir çalışma yürütülerek Tablo 19'da gösterilen 25 şaire ait toplam 7148 esere ulaşılarak veri seti oluşturulmuştur. Çalışma kapsamında çapraz doğruluma kullanıldığı için eğitim ve test veri seti ayrılmamıştır. Bütün veriler dönüşümlü olarak hem eğitim içim hem de test için kullanılmıştır. Eğitim veri seti sadece söz konusu 25 şaire ait eserler içerdiğinden çalışma kapsamında oluşturulan şair tanıma sistemi sadece ismi geçen şairler için doğru sonuç üretmektedir.

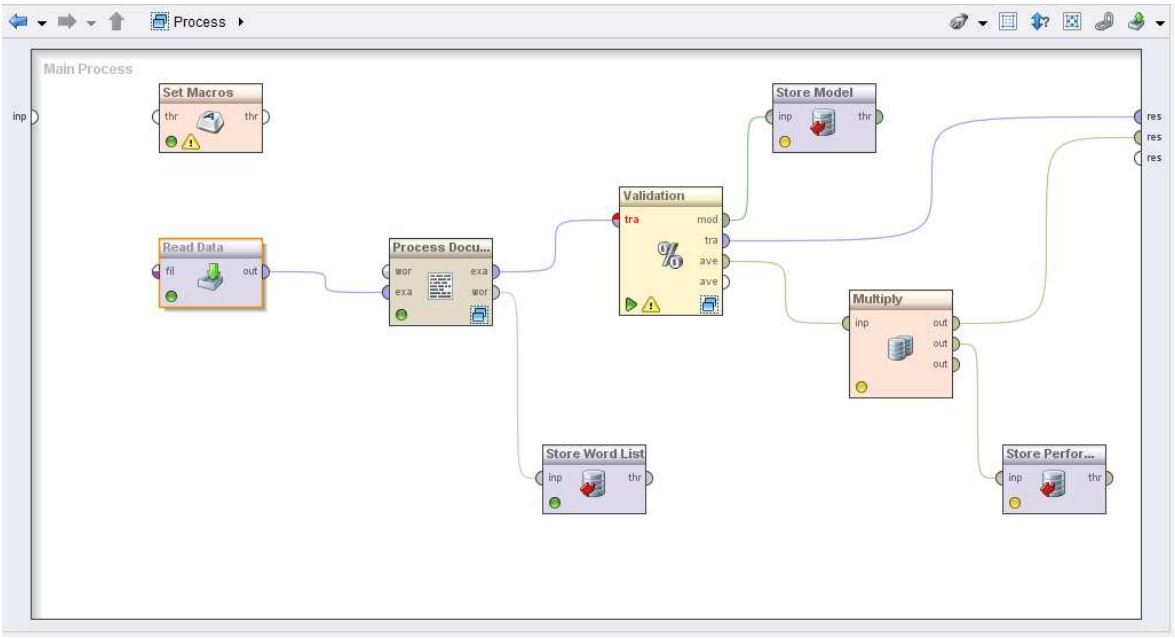
Tablo 19. Şair İsimleri ve Eser Sayıları

Şair	Sayı	Şair	Sayı
Adli Divanı	145	Cem Sultan Divanı	304
Adni Divanı	97	Dukakinzade Ahmet Divanı	295
Agah Divanı	338	Edirneli Nazmi	477
Ahmed-i Dai Divanı	303	Emri Divanı	523
Ahmet Paşa Divanı	389	Esad Divanı	270
Akif Divanı	196	Esadı Bagdadi Divanı	315
Amri Divanı	103	Fatin Divanı	258
Aşık Çelebi Divanı	161	Fedayi Divanı	378
Avni Divanı	90	Fuzuli Divanı	340
Aziz Mahmud Hüdayi Divanı	206	Nedim Divanı	390
Baki Divanı	431	Şeyh Galip	448
Bosnalı Sabit Divanı	374	Zati Divanı	205
Bursalı Talip Divanı	112	Toplam	7148

Yazar tanıma işlemi gerçekleştirilmeden önce ilk olarak iki şair seçilerek eserleri karşılaştırılmıştı. Bu karşılaştırma neticesinde %93,92 gibi yüksek bir başarı oranına ulaşıldığı için 25 şair için sözcük tabanlı yazar tanıma sistemi gerçekleştirilmiştir. Yazar tanıma sistemi için kurulacak modelin başarı oranını yükseltmek için 20 farklı model

kurulmuştur. Bu modellere ilişkin gerçekleştirilen işlemler çalışmanın 2. bölümü olan "Yapılan Çalışmalar" kısmında anlatılmıştır. Modellerin karşılaştırılması doğruluk, kappa katsayısı, duyarlılık, hassasiyet ve f-ölçütü performans değerlendirme ölçekleri kullanılarak gerçekleştirilmiştir.

Kurulan modeller neticesinde en başarılı model tespit edilerek yazar tanıma işlemi için model kurulumu tamamlanmıştır. Bunun için yapılan işlemler Rapidminer 5.3 programı aracılığıyla gerçekleştirilmiş olup ana hatları Şekil 5'de gösterilmiştir.



Şekil 5. Rapidminer'da Gerçekleştirilen Ana İşlemler

Şekil 5'de ana adımları gösterilen modelde ilk olarak dosyaların bulunduğu konumları belirlemek için "Set Macros" ile makro oluşturulmuştur. Daha sonra CSV formatına çevrilmiş veriler "Read CSV" işlemi ile programın içerisine aktarılmıştır. Bu veriler "Process Documents from Data" işlemi ile etkisiz kelimelerin çıkarılması, kelime grupları, karakter analizi ve gövdeleme gibi ön işlemler ve VU ağırlıklandırma yöntemi işlemlerine tabi tutulmuştur. Bu işlem sonucunda kelimelerine kadar ayrıştırılan veri seti "Store Word List" işlemi ile yeni şiir girileceği zaman kullanılmak amacıyla kaydedilmiştir. Ardından "Validation" işlemi ile doğrulama, sınıflandırma ve değerlendirme süreçleri gerçekleştirilmiştir. Bu işlemden sonra yeni veri girişinde kullanılmak üzere model ve performans sonuçları "Store Model" ve "Store Performance" işlemleri aracılığıyla saklanarak süreç son bulmuştur.

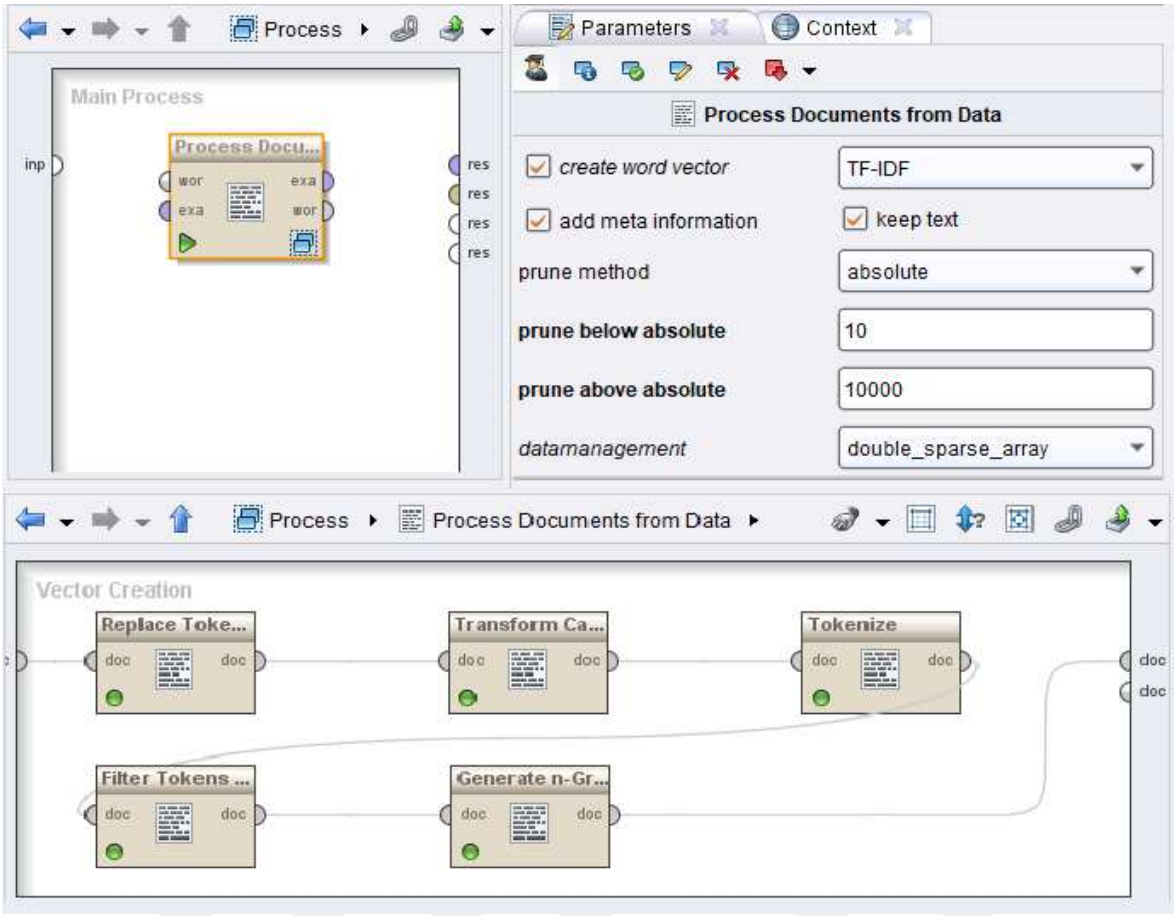
Ana hatları ile gösterilen modelin ayrıntılı çıktısı ek-1'de gösterilmiştir. Söz konusu modelin kurulumu sırasında yapılan çalışmalar neticesinde kullanılmasına karar verilen ön işlemler, ağırlıklandırma ve doğrulama yöntemleri ile sınıflandırma analizi belirlenip aşağıda anlatılmıştır.

3.1. Ön İşlemler ve Ağırlıklandırma Yöntemi

Çalışma kapsamına İlk olarak dokümanlar sisteme eklenip standartlaştırılarak kelimelerine kadar ayrıştırılmıştır. Sonrasında modeli yanlış yönlendirecek etkisiz kelimeler çıkarılarak kelime grubu işlemleri gerçekleştirilmiştir. Son olarak filtrelenecek kelime sayısı ve ağırlıklandırma yöntemi belirlendikten sonra ön işlemler ile ağırlıklandırma yöntemleri tamamlanmıştır. Yapılan çalışmalar neticesinde yazım hatalarının tespiti, gövdeleme ve karakter analizlerinin modele katkısı istenilen düzeyde gerçekleşmediğinden sistemde kullanılmamıştır. Gerçekleştirilen ön işlemeler ve ağırlıklandırma yönteminin Rapidminer programı üzerinde gerçekleştirilen işlemlerin hepsi Şekil 6'de gösterilmiştir.

Şekil 6'da ilk olarak şairlere ait PDF formatında tutulan veriler birleştirilerek sistematik bir format olan CSV formatına dönüştürülmüştür. Bu işlem PDF dosyasındaki verilerin sistematik olmamasından dolayı otomatik bir şekilde yapılamamış ve oldukça zaman almıştır. Dokümanlar oluşturulurken her bir divan şiiri bir doküman olarak belirlenmiştir. Bu sebepten toplam 7148 divan şiiri eğitim veri seti olarak değerlendirilmiştir.

CSV dosyası oluştuktan sonra ön işlemler "Process Documents from Data" düğümünün alt işlemlerinde gerçekleştirilmiştir. Sadece filtrelenecek kelime sayısının belirlenmesi ağırlıklandırma yöntemi ile beraber ilgili düğümün kendi özellikleri içerisinde belirlenmiştir.



Şekil 6. Kullanılan Ön İşlemler

Verileri standartlaştırma için ilk olarak alt işlemler içerisinden "Replace Tokes" işlemiyle veri setinde bulunan bütün "I" harfleri "ı" harfine ve "İ" harfleri "i" harfine dönüştürülmüştür. Bu işlemden sonra "Transform Cases" işlemi ile bütün karakterler küçük harfe dönüştürülmüştür. Dil çalışmalarında büyük ve küçük harfler farklı karakter olarak algılandığından bu düzeltmenin yapılması bir elzem olmuştur.

Harf dönüşümleri gerçekleştirildikten sonra "Tokonize" işlemiyle veri setinde harfler dışında bulunan rakam, noktalama işareti gibi bütün karakterler çıkarılarak veri seti kelimelerine kadar ayrıştırılmıştır. Sözcük tabanlı yazar tanıma işlemi gerçekleştirileceği için rakam ve noktalama işaretleri veri setini hem şişirmekte hem de yanlış yönlendirmektedir. Bu sebeple çıkarılmışlarıdır. Ayrıca model kelimelerin sıklıklarına dayalı kurulacağı için veri seti kelimelerine kadar ayrıştırılmıştır.

Bir sonraki işlem olan etkisiz kelimelerin çıkarılması için "Filter Tokens (by Length)" özelliği kullanılmıştır. Veri seti divan edebiyatına özel kelimeler içerdiğinden literatürde Türkçe için kullanılan etkisiz kelime listelerinin kullanılması, modeli yanlış etkileyebileceği

için tercih edilmemiştir. Bu sebepten etkisiz kelimeler için herhangi bir liste oluşturulmamış sadece kelime uzunluğu 3 harften küçük olanlar etkisiz kelime olarak belirlenip sistemden çıkarılmıştır.

Etkisiz kelimeler çıkarıldıktan sonra kelime gruplarının analizi gerçekleştirilmiştir. Programın "Generate n-gram (terms)" işlemi ile gerçekleştirilen bu analizde en başarılı sonuçlara erişmek için çeşitli denemeler gerçekleştirilmiştir. Bu çalışmalar neticesinde 2-gram yönteminin kullanıldığı durumda model en başarılı sonuçları sağlamıştır. Bu sebepten modelde 2-gram yöntemi kullanılmıştır. Bu işlemin tamamlanması ile beraber "Process Documents from Data" düğümünün alt işlemleri tamamlanmış olur.

Alt işlemlerin tamamlanması ile düğümün kendi özelliklerinden birisi olan "Prune Method" ile filtreleme yöntemi belirlenmiştir. Filtreleme yöntemi sabit bir sıklıktan az sıklık değerine sahip kelimelerin filtrelenmesi için "Absolute" olarak belirlenmiştir. Daha sonra filtrelenecek kelime sayısını belirlemek için "Prune Below Absolute" işlemi kullanılmıştır. Burada yapılan çalışmalar neticesinde veri seti için en uygun filtreleme sayısı 10 olarak belirlendiği için sıklık sayısı 10'dan az olan kelimeler modelden çıkarılmıştır.

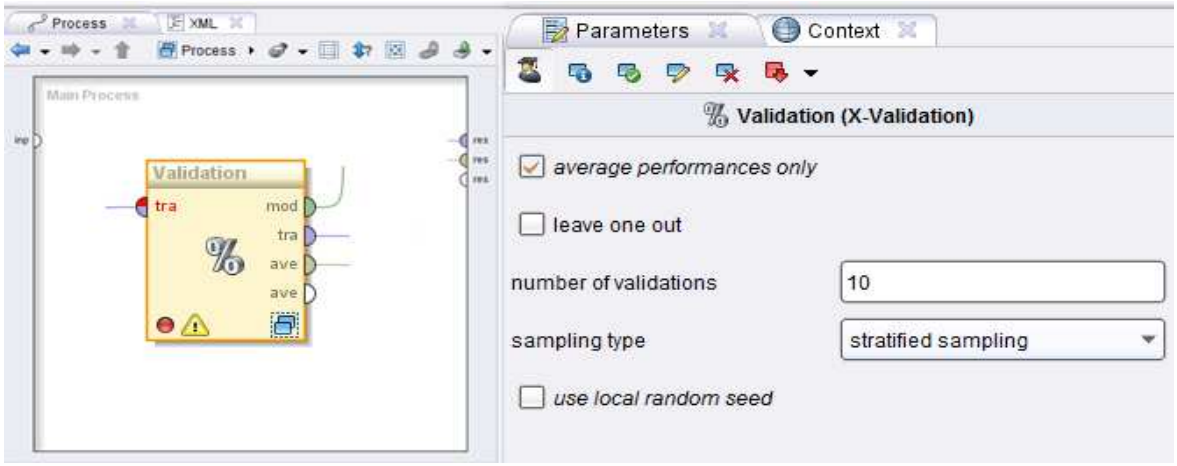
Son olarak "Process Documents from Data" işleminin "Vector Creation" parametresi ile ağırlıklandırma yöntemi gerçekleştirilmiştir. Yapılan çalışmalar neticesinde ağırlıklandırma için VU yönteminin KÇ yöntemine göre başarı oranı daha yüksek çıkmıştır. Bu nedenle kurulan modelde VU yöntemi kullanılmıştır. VU yönteminde terimi metin içerisinde ifade etmenin TS, TDS ve Normalleştirme özelliklerinin farklılaşmasına göre birçok farklı yolu bulunmaktadır. Gerçekleştirilen testler neticesinde VU yöntemi için kurulan vektörün özellikleri;

- Tablo 4'de belirtilen TS metotlarından ikinci sıradaki "f yöntemi",
- Tablo 5'de belirtilen TDS yöntemlerinden "t yöntemi",
- Tablo 6'da belirtilen normalleştirme yöntemlerinden "t yöntemi",

şeklinde belirlenmiştir.

3.2. Doğrulama Yöntemi ve Sınıflandırma Analizi

Tez çalışması kapsamında kullanılan doğrulama yöntemi, Şekil 7'de gösterildiği üzere 10'lu çapraz doğrulama yöntemidir. Burada seçilen "X-Validation" yöntemi çapraz doğrulamayı gösterirken "number of validation" seçeneği de doğrulama sayısını göstermektedir.

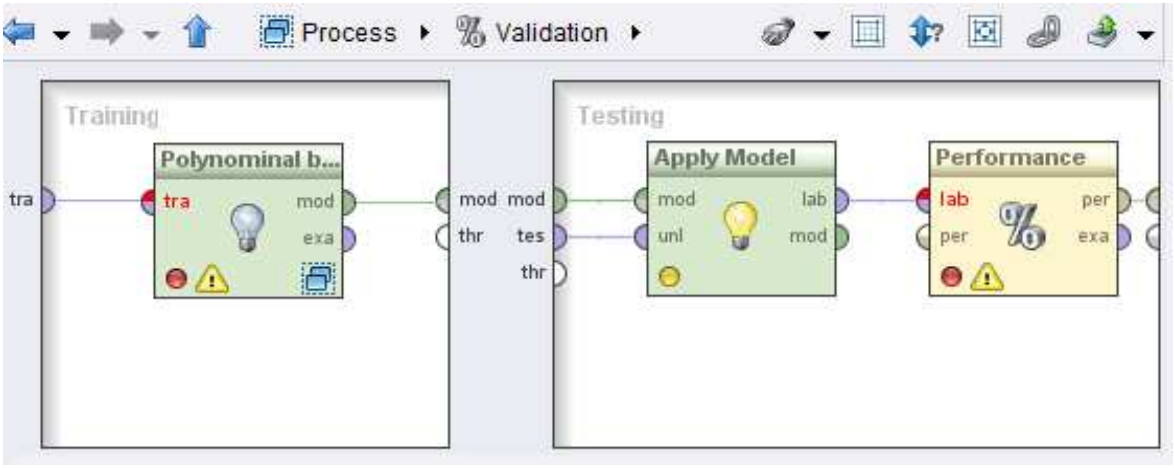


Şekil 7. Çapraz Doğrulama Yönteminin Kullanımı

Ayırma yöntemi kullanıldığında modeli ezberlemeye yol açabileceği ve yapılan testler neticesinde performansı düşük olduğu için tercih edilmemiştir. Çapraz doğrulamada ise model kurulumu ayırma yöntemine göre daha yavaş gerçekleşecek olmasına rağmen herhangi bir zaman kısıtı olmadığı için bu sorun önem teşkil etmemiştir. Çapraz doğrulama için 10'lu doğrulamanın optimum olduğu belirlenmiş ve analizler 10'lu çapraz doğrulama yöntemi kullanılarak gerçekleştirilmiştir.

Çalışma kapsamında kullanılan sınıflandırma yöntemini belirlemek için birçok farklı model kurularak başarıları test edilmiştir. Bunlar KBÖ ve ÖUÖ ölçüm değerleri ile KEK, NB ve onun özelleşmiş durumu olan ÇNB, KA, LR ve DVM yöntemleridir. Bu yöntemler içerisinde en başarılı sonuçlara DVM ile erişildiği için sınıflandırma işlemi DVM yöntemi kullanılarak gerçekleştirilmiştir.

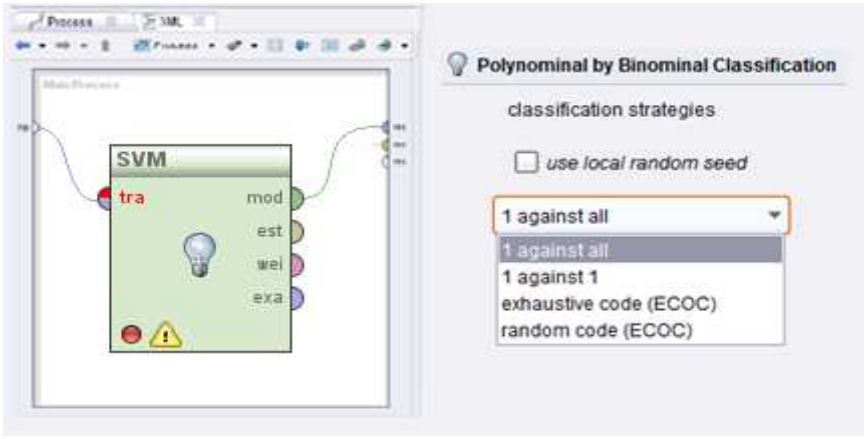
Rapidminer programında sınıflandırma ve değerlendirme işlemleri Şekil 7'de gösterilen "Validation" düğümünün alt işlemlerinde gerçekleşmektedir. Bunun nedeni sınıflandırma işlemiyle doğrulamanın ayrı bir şekilde düşünülemeyecek olmasıdır. Bu sürece ait işlemler Şekil 8'de gösterilmiştir.



Şekil 8. Sınıflandırma Analizleri

Veri seti Şekil 8'deki iş akışında "Training" bölümünde eğitime, "Testing" kısmında teste tabi tutulmuştur. Eğitim bölümünde "Polynominal by Binominal Classification" işlemi ile beraber DVM sınıflandırması gerçekleştirilmiştir. DVM yöntemi kendi algoritması gereği iki sınıflıdır. Ancak veri setinde 25 şair bulunduğu için veri seti ikiden fazla yani çok sınıflıdır. Bu durumu ortadan kaldırmak için Şekil 9'da gösterildiği üzere Rapidminer'ın "Polynominal by Binominal Classification" işleminden yararlanılır.

Test aşamasında ilk olarak "Apply Model" işlemi ile eğitilerek modeli kurulan veriler test verileri ile teste tabi tutulurlar. Ancak bu verileri sadece test etmek yeterli değildir. Test edilen verilerin başarılarını da gözlemlemek veya kayıt altına almak gereklidir. Bu sebepten "Performance" işlemi ile modele ilişkin değerlendirme ölçekleri belirlenir. Ancak doğrulama için 10-lu çapraz doğrulama kullanıldığı için bu işlemler 10 defa tekrarlanmıştır. Her bir modele ait değerlendirme sonuçlarının aritmetik ortalaması genel değerlendirme sonuçlarını oluşturmuştur.

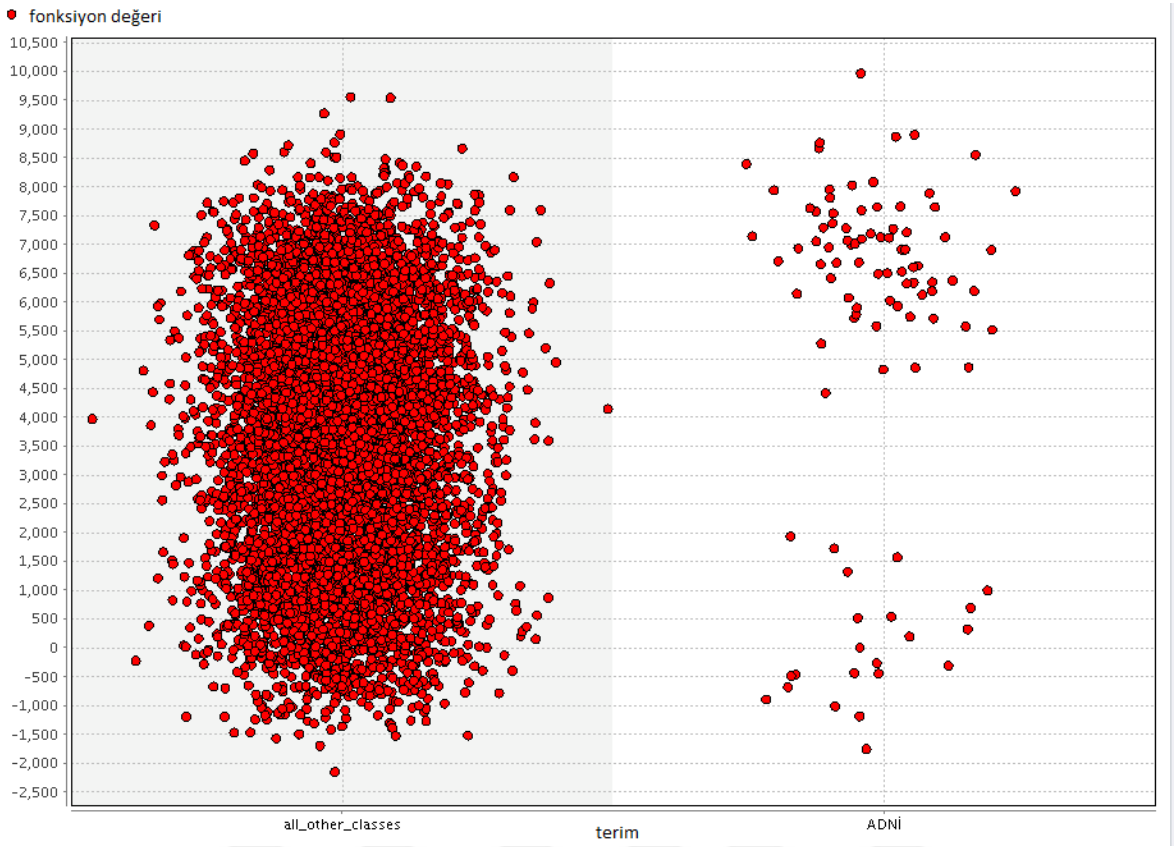


Şekil 9. DVM Kullanımı

DVM yöntemini çok terimli sınıflandırmada kullanmak için her bir terime karşılık diğer bütün terimleri sınıflandırmak gerekir. Ancak bu durumda Şekil 10'da gösterildiği gibi her bir sınıf için ayrı sınıflandırma gerçekleştirilir. Sonuç olarak sınıflandırma sonuçları birleştirilerek DVM sınıflandırıcısı oluşturulur.

Şekil 10. DVM Sınıflandırıcı Karşılaştırmaları

DVM sınıflandırıcısı içerisindeki karşılaştırmalara örnek teşkil etmesi için ilk incelenen divan olan Adni Divanı ile diğer divanların karşılaştırma sonuçlarını içeren saçılım grafiği Şekil 11'de gösterilmiştir. Grafikten de anlaşılacağı üzere verilerin büyük çoğunluğu diğer divanlarda yoğunlaşmıştır. Ancak Adni Divanı'nda bulunan terimler sayıca az olduğundan sınıflandırıcı için önem düzeyi yüksektir. Bu hesaplamalar neticesinde DVM için sınıflandırma işlemi gerçekleştirilmiştir.



Şekil 11. Adni-Diğerleri Saçılım Grafiği

Model kurulumu, sınıflandırma analizlerinden sonra değerlendirme ölçütlerinin hesaplanması ile son bulmuştur. Kurulan modelde kullanılan yöntemleri gösteren meta veriler Tablo 20'de gösterilmiştir.

Tablo 20. Kurulan Modelin Meta Verileri

Etkisiz Kelimelerin Çıkarılması	Gövdeleme	Kelime Grupları	Karakter Analizi	Kelime Filtreleme	Ağırlıklan-dırma	Sınıflandır-ma
Kelime Uzunluğu 3'den Kısa	Sıfır Gövdeleme	2-gram	Yok	10	VU	DM

Yazar tanıma için araştırılan özelliklerin başarı oranını maksimize etmek için 20 farklı model kurulmuş ve en yüksek başarı oranına Tablo 20'de gösterilen parametre değerleri neticesinde ulaşılmıştır. Bu bilgiler ışığında gerçekleştirilen modelde 3 harften kısa

kelimeler etkisiz kelime olarak belirlenerek çıkarılmış, sıfır gövdeleme uygulanmış, 2li kelime grubu kullanılmış ve karakter analizi gerçekleştirilmemiştir. Daha sonra 10 kelimeden az sıklığa sahip kelimeler filtrelenerek VU ile ağırlıklandırılıp DM ile sınıflandırma gerçekleştirilmiştir. Ayrıca 10'lu çapraz doğrulama yöntemi kullanılmıştır.

Bu bilgiler ışığında kurulan modele ilişkin değerlendirme sonuçları Tablo 21'de yer almaktadır.

Tablo 21. Modelin Değerlendirme Sonuçları

Yöntem	Doğruluk	Kappa	Duyarlılık	Hassasiyet	F Ölçütü
Sonuçlar	91,45	0,910	88,13	92,44	90,23

Tablo 21'de gösterilen sonuçlara göre modelde %91,45'lik doğruluk ölçütüne 0,910 kappa değerine, %88,13 duyarlılık, %92,44 hassasiyet ve %90,23 f-ölçütü değerine ulaşılmıştır. Buradan yola çıkarak kurulan modelin mevcut yazarlara ait bir divan şiirinin şairini %91,45 olasılıkla doğru tahmin edeceğini söylemek mümkündür.

3.3. Divan Şairlerini Tanımada Etkili Kelimeler

Yazar tanıma işlemi için kurulan model neticesinde edebi açıdan önemli bazı çıktılara ulaşılmıştır. Bu çıktılar her bir şairin belirlenmesini istatistiksel olarak fazla etkileyen önemli kelimelerdir. Kurulan son modelde ikili kelime gruplarının veri setinde bulunması anlamlı bulunduğu için kelime listelerinde istatistiksel olarak önemli bulunan kelimelerin yanında ikili kelime grupları da yer almaktadır.

Veri setinde bulunan bütün kelimelerin her bir yazar için ağırlık değeri oluşmasına rağmen bazı kelimelerin ağırlık değerleri diğerlerine göre daha yüksektir. Söz konusu yüksek değerler ilgili şairin seçimini olumlu yönde etkileyen kelimelerdir. Bahsi geçen önemli kelime listeleri her bir yazar için ayrı bir şekilde Tablo 22-46'da gösterilmiştir.

Tablo 22-46'da gösterilen bütün divanlar için Divan Edebiyatında mahlas denilen şairin isminin en anlamlı kelime olduğu göze çarpmaktadır. Onun dışında kalan önemeli kelimeler ise şairlere göre farklılıklar göstermektedir.

Tablo 22. Adli Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
adli	hicr	milk	yeg
fikri muhal	her gece	fülfül	enis
rakib	hırka	kuy	firak
munis	sufi	lulu	gamgin
dastan	berk		

İlk etkili kelimeler Adli Divanı için hazırlanmış olup Tablo 22'de gösterilmiştir. Söz konusu önemli kelimelere göre "hicr", "milk" ve "yeg" olumlu yönde en belirleyici kelimelerdir. Ayrıca "fikri muhal" ve "her gece" tamlamaları da beraber kullanılan önemli kelime gruplarıdır.

Tablo 23. Adni Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
adni	bigi	yüz	lale
milk	çeşme	gamze	yaş
tan degül	cennet içre	zülüf	hadd
boy	sen	can	ışk

Avni Divanı'ndan sonra en az eser sayısına sahip Adni Divanı'nda Tablo 23'de gösterildiği üzere toplam 16 adet önemli kelime tespit edilmiştir. Bu önemli kelimeler içerisinde "bigi", "yüz" ve "lale" kelimeleri olumlu yönde en belirleyici kelimeler olarak tespit edilmiştir. Öte yandan "tan degül" ve "cennet içre" ikili kelime grupları da beraber kullanılan önemli kelime gruplarıdır.

Tablo 24. Agah Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
agah	şüh	gamze	hün
süz	hurşid	arzü	büy
dehr	mamür	cüşı	hamuş

Çalışma kapsamında incelenen eser sayısı bakımından fazla esere sahip şairlerden birisi olan Agah Divanı için Tablo 24'de gösterilen önemli kelimelere göre "şüh", "gamze" ve "hün" kelimeleri olumlu yönde en belirleyici kelimeler olarak belirlenmiştir. Buna karşın Agah Divanı için hiçbir tamlama istatistiksel olarak önem teşkil etmemiştir.

Tablo 25. Ahmed-i Dai Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
dai	gonçe	cazu	müşk
leb	incü	ahd	maşuk
badı saba	bade	abı hayat	ruzigar
hüsn letafet			

Yaklaşık 300 eserinin incelendiği Ahmed-i Dai Divanı için toplam 13 adet önemli kelime belirlenmiş ve Tablo 25'de gösterilmiştir. Söz konusu önemli kelimelerden "gonçe", "cazu" ve "müşk" kelimeleri olumlu yönde en belirleyici kelimeler olarak karşımıza çıkmaktadır. Bu divan için "badı saba", "abı hayat" ve " hüsn letafet" tamlamaları da yüksek derecede önem teşkil eden kelime gruplarıdır. Tepsit edilen 13 önemli kelimeden 3 tanesinin tamlama olması açısından Ahmed-i Dai'nin sıklıkla tamlama kullandığı öngörülebilir. Ancak bunun kesin olarak belirlenmesi için daha kapsamlı bir çalışmanın gerçekleştirilmesi gerekmektedir.

Tablo 26. Ahmet Paşa Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
ahmed	zülüf	doğru	yüz
ahmed gibi	cemal	kaş	leb
her gece	rakib	hüsn	ıyd

Çalışma kapsamında toplam 389 eseri incelenen Ahmet Paşa Divanı'nda Tablo 26'da gösterildiği üzere 12 kelime önemli olarak kabul edilmiştir. Bu kelimeler içerisinden "zülüf", "doğru" ve "yüz" olumlu yönde en belirleyici kelimelerdir. İkili kelime grupları incelendiğinde şairin ismi olan "ahmed" kelimesi ile "gibi" kelimesinin beraber kullanıldığı "ahmed gibi" ikili kelime grubunun önemli olduğu ortaya çıkmıştır. Buradan şairin kendinden bahsederken sıklıkla "ahmed gibi" ifadesini kullandığı anlaşılmaktadır. Onun dışında "her gece" kelime grubu da önem teşkil eden bir başka kelime grubu olmuştur.

Tablo 27. Akif Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
akif	hatır	muhlis	peyrevlik
gece	zülal	cenab	firkat
hayal	kuşe	hazret	nazım

Akif Divanı için gerçekleştirilen analizler neticesinde Tablo 27'de gösterildiği üzere toplam 12 kelime önemli olarak belirlenmiştir. Bu önemli kelimeler içerisinden "hatır", "muhlis" ve "peyrevlik" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan ilgili divan için herhangi bir kelime grubu önem teşkil etmemiştir.

Tablo 28. Amri Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
amri	bir güzel	buse	müşkin
sim	kafur	ciger	büt
gamze	namus	müşevveş	lale gibi

Amri Divanı için Tablo 28'de gösterilen önemli kelimelere göre "buse", "müşkin" ve "sim" kelimeleri olumlu yönde en belirleyici kelimelerdir. Bu divan için "bir güzel" ve "lale gibi" tamlamaları büyük derecede önem teşkil eden kelime grupları olmuştur.

Tablo 29. Aşık Çelebi Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
aşık	ıyş	enduh	tig
himmet	mesel	devlet	askeri
peyman	el	reyhan	gerdun
surahi	ilahi	kangı	nişan

Aşık Çelebi Divanı için Tablo 29'da gösterildiği üzere toplam 16 adet önemli kelime tespit edilmiştir. Bu kelimelere göre "ıyş", "enduh" ve "tig" olumlu yönde en belirleyici kelimelerdir. Dikkat edilecek bir başka husus "devlet", "askeri" ve "ilahi" gibi kelimeler Aşık Çelebi'nin eserlerinin konularının belirlenmesi için önemli kelimeler olarak karşımıza çıkmıştır. Diğer taraftan bu divan için herhangi bir kelime grubu önem teşkil etmemiştir.

Tablo 30. Avni Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
avni	hançer	allah	hatır
mar	müje	üryan	revnak
zevk	nuş	hecr	emir
gamze	firdevs	geda	nüzul
dilber	gül		

Çalışma kapsamında en az eser sayısına sahip Avni Divanı için toplam 18 önemli kelime tespit edilmiş olup Tablo 30'da gösterilmiştir. Söz konusu önemli kelimelere göre Avni Divanı için "hançer", "allah" ve "hatır" kelimeleri olumlu yönde en belirleyici kelimeler olarak karşımıza çıkmıştır. Öte yandan bu divan için herhangi bir kelime grubu önem teşkil etmemiştir.

Tablo 31. Aziz Mahmud Hüdayi Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
hüda	mevla	meded	allah
esirgemek	varlık	katre	ihsan
tevhid	nefis	cenab	kul
sultan	bülbül	gaflet	

Aziz Mahmud Hüdayi Divanı için Tablo 31'de gösterilen önemli kelimelere göre "mevla", "meded" ve "allah" kelimeleri olumlu yönde en belirleyici kelimelerdir.

Tablo 32. Baki Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
baki	aşk	sakı	mürg
lülu	gam	gonca	nihal
cam	ney	sultan murad	hvace
çerag	visal	husrev	nevbahar
kaşki	nükte	el	

Tablo 32'de gösterilen önemli kelimelere göre Baki Divanı için "aşk", "sakı" ve "mürg" kelimeleri olumlu yönde en belirleyici kelimelerdir. Bu divan için "sultan murad" kelime grubu büyük derecede önem teşkil eden tamlamadır.

Tablo 33. Bosnalı Sabit Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
sabit	sultan	kumaş	pir
safha	şahinşeh	sadır	harun
suzan	ihlas	kilk	biz
gülsen	derun	maraz	pare
ilaç			

Tablo 33'de gösterilen önemli kelimelere göre Bosnalı Sabit Divanı için "sultan", "kumaş" ve "pir" kelimeleri olumlu yönde en belirleyici kelimelerdir. Bu divan için hiçbir tamlama önem teşkil etmemiştir.

Tablo 34. Bursalı Talip Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
talib	gitmemek	avaz	hayal
yusuf	gam	gül	suhan
mevc	siyeh	arzu	müşkil
şahbaz	neşve	aguş	küşade
semend	perde		

Tablo 34'de gösterilen önemli kelimelere göre Bursalı Talip Divanı için "gitmemek", "avaz" ve "hayal" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan ilgili divan için hiçbir tamlama önem teşkil etmemiştir.

Tablo 35. Cem Sultan Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
cem	herdem	gem	küy
sanem	vakit	gamzen okı	hatun
dôst	derd	hayal	ışk
zülüf	müşg	leb	yad
cadü	gamze		

Tablo 35'de gösterilen önemli kelimelere göre "herdem", "gem" ve "küy" kelimeleri olumlu yönde en belirleyici kelimelerdir. Bu divan için "gamzen okı" ikili kelime grubu büyük derecede önem teşkil eden tamlamadır.

Tablo 36. Dukakinzade Ahmet Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
ahmed	dost	cah	habib
vahdet	cehd etmek	dilber	saç
zahid	hüda	guş	cehd
goncaleb	sufi	kati	hüseyn

Tablo 36'da gösterilen önemli kelimelere göre "dost", "cah" ve "habib" kelimeleri olumlu yönde en belirleyici kelimelerdir. Bu divan için "cehd etmek" kelime grubu önem teşkil eden ikili kelime grubudur.

Tablo 37. Edirneli Nazmi Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
dust	nazım	ana	afitab
sürur	ejdeha	vedduha	muhibbi
saba	süha	vefa	hecr
revani	mehlika	son	dilara
şeyda			

Tablo 37'de gösterilen önemli kelimelere göre "dust", "ana" ve "afitab" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan ilgili divan için hiçbir tamlama önem teşkil etmemiştir.

Tablo 38. Emri Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
emir	gami	hatt	mahi
ahu	afitabi	murg	rah
kaş	üst	yusuf	dehani
serab	gam	tıfil	

Çalışma kapsamında 523 eseri incelenerek en fazla eser sayısına sahip Emri Divanı için Tablo 38'de gösterildiği üzere toplam 15 önemli kelime tespit edilmiştir. Söz konusu önemli kelimeler içerisinden "gami", "hatt" ve "mahi" kelimeleri olumlu yönde en belirleyici olanlarıdır. Buna karşın ilgili divan için herhangi bir ikili kelime grubu önem teşkil etmemiştir.

Tablo 39. Esad Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
esad	ruy	gonca	pürtab
tab	yar	mecaz	mihir
ayan	humar	tig	hecr
tahsil	fikr	nigah	ahir

Tablo 39'da gösterilen önemli kelimelere göre "ruy", "gonca" ve "pürtab" kelimeleri olumlu yönde en belirleyici kelimelerdir. Bu divan için hiçbir tamlama büyük derecede önem teşkil etmemiştir.

Tablo 40. Esad-ı Bagdadi Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
esad	mürg	lal	cuş
gönül	rüy	civan	gonce
zincir	küşe	müy	kadı
aşina	güş	zan	pay
nüş	takat	reng	

Tablo 40'da gösterilen önemli kelimelere göre "mürg", "lal" ve "cuş" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan bu divan için hiçbir tamlama büyük derecede önem teşkil etmemiştir.

Tablo 41. Fatin Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
fatin	hasret	efendi	rical
alem	abdülmecid	akl	gaddar
tig	biçare	allah	leyla
peyrev	togrı	şeh	tab
iltifat			

Tablo 41'de gösterilen önemli kelimelere göre "hasret", "efendi" ve "rical" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan bu divan için hiçbir tamlama büyük derecede önem teşkil etmemiştir.

Tablo 42. Fedayi Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
fida	bigi	can	kol
lokman	istemek	her nazar	geh geh
müsülman	şübhesüz	dilemek	kadem
degül	aynı	onsuz	cerh

Tablo 42'de gösterilen Fedayi Divanı için önemli kelimelere göre "bigi", "can" ve "kol" olumlu yönde en belirleyici kelimelerdir. Bu divan için "her nazar" ve "geh geh" ikili kelime grubu büyük derecede önem teşkil etmiştir.

Tablo 43. Fuzuli Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
fuzuli	ciğer	terk	dağ
tiğ	gün	gönül	vehm
izhar	aşk	ahval	meğer
mikdar	tiğ	mutlak	şarabi

Fuzuli Divanı için Tablo 43'de gösterilen önemli kelimelere göre "ciğer", "terk" ve "dağ" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan bu divan için hiçbir tamlama büyük derecede önem teşkil etmemiştir.

Tablo 44. Nedim Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
nedim	ser	sadır	dil ber
bad	kilk	asaf	gerden
hamdillah	leb riz	habib	pay
dil keş	fasl	guşe	dükan
düstur	tiğ	gurre	ruy

Tablo 44'de gösterilen Nedim Divanı için önemli kelimelere göre "ser", "sadır" ve "bad" olumlu yönde en belirleyici kelimelerdir. Öte yandan "dil ber" ve "leb riz" ikili kelime grupları büyük derecede önem teşkil etmiştir.

Tablo 45. Şeyh Galip Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
galib	suhan	hoş	ehl
şule	çeşm	hazret	sırr
kuşe	gonce	hun	mahtab
ateş	gevher	zehir	şems
tig	tar	tavır	harim
suziş	güruh		

Tablo 45'de gösterildiği üzere Şeyh Galip Divanı toplam 22 önemli kelime ile en fazla önemli kelimeye sahip divan olmuştur. Söz konusu önemli kelimeler içerisinden "suhan", "hoş" ve "ehl" olumlu yönde en belirleyici kelimeler olarak tespit edilmiştir. Öte yandan Şeyh Galip Divanı için hiçbir kelime grubu büyük derecede önem teşkil etmemiştir.

Tablo 46. Zati Divanı Önemli Kelime Listesi

Kelime	Kelime	Kelime	Kelime
zati	çerh	padşeh	feth
asman	şevkle	aftab	kalb
sultan süleyman	taht	kasr	server
padşah	lamekan	arş	hilal
lutf	rifat	subhdem	

Çalışma kapsamında incelenen son divan olan Zati Divanı'na ait önemli kelimeler Tablo 46'da gösterilmiştir. Söz konusu önemli kelimelere göre "çerh", "padşeh" ve "feth" kelimeleri olumlu yönde en belirleyici kelimelerdir. Öte yandan bu divan için "sultan süleyman" ikili kelime grubu büyük derecede önem teşkil etmiştir.

4. SONUÇLAR VE ÖNERİLER

Bu çalışma ile divan edebiyatında kelime odaklı bir metin sınıflandırma çalışması gerçekleştirilmiştir. Başka bir ifade ile divan edebiyatı eserlerine ilişkin yazar tanıma sistemi kurulmuştur. Bu sistem kullanılarak modele girdi olarak yeni bir eser dâhil edildiğinde ilgili eserin yazarına ilişkin istatistiksel bir tahmin gerçekleşmektedir. Yapılan analizler neticesinde kurulan model %91,45 doğruluk ve %90,23 f-ölçütü değerleriyle tahmin gerçekleştirmiştir.

Şair tanıma işlemi dışında gerçekleştirilen analizler neticesinde çalışma kapsamında araştırılan 25 şair için ayırt edici kelimeler tespit edilmiştir. Divan Edebiyatı açısından şairin karakteristiğini gösteren bu kelimeler oldukça önem teşkil etmektedir.

Şair belirleme işleminde şairin kendi ismi ya da isminden türetilmiş kelimeler seçimi olumlu yönde etkileyen istisnai bir durum olmuştur. Bu sebepten söz konusu veriler çıkarılıp bir çalışma gerçekleştirilmiş olup %83,17 doğruluk ve %79,60 f-ölçütü değerlerine ulaşılmıştır. Buradan yola çıkarak şair isimlerinin çıkarılmasının modelin başarısını olumsuz etkilediği sonucuna varılabilir. Ancak tahmin edilecek yeni şiirler de şairin ismini içerebildiğinden bu değerler modelden çıkarılmamıştır.

Bu çalışmada kelime odaklı yaklaşım ile divan edebiyatı eserleri için metin sınıflandırma işleminin başarılı sonuçlar verdiği gösterilmiştir. İlerleyen süreçte yeni özellikler ve yaratıcı düşünceler ile bu konuyu geliştirecek çok sayıda çalışmanın yapılabileceği öngörülmektedir.

İleride içerisinde aruz vezni, kafiye, redif, uyak, beyitteki sesli ve sessiz harf sayıları, beyitin uzunluğu ve eserdeki toplam beyit sayısı, kullanılan tamlama sayısı gibi farklı özellik vektörlerini barındıran çalışmaların yapılması konu için çok faydalı olacaktır. Bu sayede kurulacak yeni modellerle daha başarılı tahminlerin yapılabileceği öngörülmektedir. Üstelik bu özellikler vektörlerinin ayrı ayrı incelenmesi neticesinde divan edebiyatına ilişkin önceden bilinmeyen bazı yeni yapısal özelliklerin de keşfedilebileceği düşünülmektedir.

Divan edebiyatını bütünüyle kapsayacak bir çalışma için literatürdeki bütün divan şairlerinin eserlerini içeren bir çalışma gerçekleştirilebilir. Gerçekleştirilecek yeni çalışmayla divan edebiyatının temel öğelerini içeren bir kelime vektörü hazırlanarak bir eserin divan edebiyatına ait olup olmadığı kontrol edilebilir. Kontrol sağlandıktan sonra şair tanıma dışında eserlerin bölge, dönem ve tür belirleme işlemlerine ait çalışmalar yapılabilir.

İleride gerçekleştirilecek bir çalışmayla elde edilen çıktıların boyutlarını indirgemek ve ayırt edici çıktıları tespit edebilmek için öznitelik seçimi gerçekleştirilebilir. İleriye ve geriye doğru seçilim yöntemleri kullanılarak öznitelik seçimi denenmiş olsa da kullanılan bilgisayarların donanım eksikliğinden ötürü analizler tamamlanamamıştır.

Çalışma kapsamında kullanılan sınıflandırma yöntemleri dışında YSA, Lojistik Regresyon, Diskriminant Analizi gibi farklı algoritmalar kullanılarak başarı oranları test edilebilir. Ayrıca metin madenciliği çalışmalarında başarılı ağırlıklandırma yöntemi olan Okapi BM25 yöntemi kullanılabilir.

5. KAYNAKLAR

1. Grimes, S., A Brief History of Text Analytics, http://www.b-eye-network .com/view/6311 02 Ocak 2018.

2. Manning, C., Raghavan, P. ve Schütze, H., Introduction to Information Retrieval, Cambridge University Press, Cambridge, 2008.

3. Feldman, R. ve James, S., The Text Mining Handbook, Cambridge University Press, Cambridge, 2007.

4. Pala, İ., Divan Edebiyatı, Kapı Yayınları, İstanbul, 1992.

5. Varol, M., Metin Madenciliği Yöntemlerini Kullanarak Türkçe Dokümanlarda Tür ve Yazar Tanıma, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta, 2011.

6. Akün, Ö., İslam Ansiklopedisi: Divan Edebiyatı. TDV İslam Ansiklopedisi, http://www.islamansiklopedisi.info/dia/ayrmetin.php?idno=090389 29 Ekim 2017.

7. Stamatatos, E., Fakotakis, N. ve Kokkinakis, G., Automatic Text Categorization in Terms of Genre and Author, Association for Computational Linguistics, 26,4 (2001) 471-495.

8. Gerritsen, C., Authorship Attribution Using Lexical Attraction, Master Thesis, Massachusetts Institute of Technology, 2003.

9. Mosteller, F. ve Wallace, D.L., Inference in an Authorship Problem: A Comparative Study of Discrimination Methods Applied to the Authorship of the Disputed Federalist Papers, American Statistical Association, 302 (1963).

10. Can, E.F., Can, F., Duygulu, P. ve Kalpakli, M., Automatic Categorization of Ottoman Literary Texts by Poet and Time Period, International Symposium on Computer and Information Sciences (ISCIS 2011), (2011) 51-57.

11. Gheith, M. ve El-Sadany, T., Arabic Morphological Analyzer on a Personal Computer, Proceedings of the Arabic Morphology Workshop, (1987).

12. Al-Fedaghi, S. ve Al-Anzi, F., A new algorithm to generate Arabic root-pattern forms, 1989.

13. Al-falahi, A., Ramdani, M. ve Bellafkih, M., Machine Learning for Authorship Attribution in Arabic Poetry, International Journal of Future Computer and Communication, 6,2 (2017) 2–6.

14. Mazdak, N., FarsiSum - a Persian Text Summarizer, Master Thesis, Stockholm University, Stockholm, 2004.

15. Arabsorkhi, M. ve Feili, H., Using Bayesian Model to Persian Text Classification, Second Workshop on Persian Language and Computer, (2006) 245-249.

16. Hamidi, S., Razzazi, F. ve Ghaemmaghami, M.P., Automatic meter classification in Persian poetries using Support Vector Machines, International Symposium on Signal Processing and Information Technology, (2009).

17. Çatal, C., Erbakırcı, K. ve Erenler, Y., Computer-based Authorship Attribution for Turkish Documents, Turkish Symposium on Artificial Intelligence and Neural Networks, (2003).

18. Diri, B. ve Amasyali, M.F., Automatic Author Detection for Turkish Texts, Artificial Neural Networks and Neural Information Processing, (2003) 1–4.

19. Amasyalı, M.F. ve Diri, B., Automatic Turkish Text Categorization in Terms of Author , Genre and Gender, 11th International Conference on Applications of Natural Language to Information Systems, (2006) 221–226.

20. Doğan, S., Türkçe Dokümanlar için N-gram Tabanlı Sınıflandırma: Yazar, Tür ve Cinsiyet Tanıma, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006.

21. Örücü, F., Turkish Language Characteristics and Author Identification, Master Thesis, Dokuz Eylül University, The Graduate School of Natural and Applied Sciences, İzmir, 2009.

22. Yasdi, M. ve Diri, B., Soyut Özellik Çıkarımı İle Yazar Tanıma, Sinyal işleme ve İletişim Uygulamaları, (2012) 1–4.

23. Kolyiğit, Ö., Aşlıyan, R. ve Günel, K., Türkçe Dokümanlar için Yazar Tanıma, XIV. Akademik Bilişim Konferansı, Şubat 2012, Bildiriler Kitabı, 423–428.

24. Levent, V.E. ve Diri, B., Türkçe Dokümanlarda Yapay SinirAğları ile Yazar Tanıma, XVI. Akademik Bilişim Konferansı, Şubat 2014, Bildiriler Kitabı, 735–741.

25. Kuzu, R., Author Recognition in Online Social Platforms, Master Thesis, Boğaziçi University, The Graduate School of Natural and Applied Sciences, İstanbul, 2017.

26. Bilgin, A., İmalat Sanayi Su, Atıksu ve Atık Verilerinin Veri Madenciliği Yaklaşımı ile İncelenmesi, TÜİK Uzmanlık Tezi, Türkiye İstatistik Kurumu, Trabzon, 2015.

27. Levenshtein, V., Binary Codes Capable of Correcting Deletions, Insertions, and Reversals, Soviet Physics Doklady, 10,8 (1966).

28. http://hunspell.github.io/, Hunspell, 06 Kasım 2017.

29. Akın, A., Akın, M., Zemberek , an open source NLP framework for Turkic Languages, 2007.

30. Leskovec, J., Rajaraman, A., ve Ullman, J., Mining of Massive Datasets, 73–128 Stanford University Press, 2011.

31. Lovins, J.B., Development of a Stemming Algorithm, Mechanical Translation and Computational Linguistics, (1968).

32. Porter, M.F., An algorithm for suffix stripping, 1980, 130–137.

33. Çelen Pollard, A. ve Pollard, D., Turkish a Complete Course for Beginners, Teach Yourself, 1996, 257–258.

34. Köksal, A., Tümüyle Özdevimli Deneysel Bir Belge Dizinleme ve Erişim Dizgesi, 1981.

35. Altıntaş, K. ve Can, F., Stemming for Turkish: a Comparative Evaluation, 2002, Proceedings of the 11th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN) , 181–188.

36. Dalkılıç, G. ve Çebi, Y., A 300 MB Turkish Corpus and Word Analysis, 2002, Advances in Information Systems: Second International Conference, ADVIS, Berlin: Springer-Verlag, 205–212.

37. Sever, H. ve Tonta, Y., Truncation of Content Terms for Turkish, CICLing, Mexico, 2006.

38. Can, F., Kocberber, S., Balçık, E., Kaynak, C. ve Öcalan, Çağdaş Vursavaş, O., Information retrieval on Turkish texts, Journal of the American Society for Information Science and Technology, 59 (2008) 407–421.

39. Oflazer, K., Hakkani-Tür, D. ve Tür, G., Design for a Turkish Treebank, Linguistically Interpreted Corpora: EACL Post-Conference Workshop, 1999, 1-9.

40. Eryiğit, G. ve Adalı, E., An Affix Stripping Morphological Analyzer for Turkish, IAESTED International Conference Artificial Intelligence and Applications, (2004) 299-304.

41. Alpoçak, A., Kut, A. ve Özkarahan, E., Bilgi Bulma Sistemleri için Otomatik Türkçe Dizinleme Yöntemi, Dokuz Eylül Üniversitesi Bilişim Bildirileri, 1995, 247-253

42. Jurafsky, D., and Martin, J., Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, Prentice H., New Jersey, 1999, 189–232

43. Doğan, S., Türkçe Dokümanlar için N-gram Tabanlı Sınıflandırma: Yazar, Tür ve Cinsiyet Tanıma, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2006.

44. Madsen, R.E., Sigurdsson, S., Hansen, L.K. ve Larsen, J., Pruning The Vocabulary For Better Context Recognition, Proceedings of the 17th International Conference on Pattern Recognition, 2004.

45. Kumar, L. ve Bhatia, P.K., Text Mining: Concepts, Process and Applications, Journal of Global Research in Computer Science, 4,3 (2013) 36–39.

46. Zellig, H., Distributional Structure, (1954) 146–162.

47. Şevik, U., Retina Görüntülerinin Kalite Değerlendirmesi ve Diyabetik Retinopati Hastalığının Tespiti, Doktora Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Trabzon, 2014.

48. Alpaydın, E., Introduction to Machine Learning, MIT Press, Cambridge, 2010.

49. Wang, J., Neskovic, P. ve Cooper, L., Improving Nearest Neighbor Rule With A Simple Adaptive Distance Measure, Science Direct Pattern Recognation Latters 28, (2006) 207–213.

50. Stephen, M.S., Thomas Bayes Bayesian Inference, Journal of the Royal Statistical Society, (1982) 250–258.

51. Ardıl, E., Esnek Hesaplama Yaklaşımı ile Yazılım Hata Kestirimi, Yüksek Lisans Tezi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Edirne, 2009.

52. Rennie, J.D.M., Shih, L., Teevan, J. ve Karger, D.R., Tackling the Poor Assumptions of Naive Bayes Text Classifiers, Proceedings of Twentieth International Conference on Machine Learning, 1973 (2003) 616–623.

53. Gülpınar, V., Avrupa Birliği Ülkeleri ile Türkiye'nin Ekonomik Göstergelerinin Karar Ağacı Yöntemi İle Karşılaştırılması, Yüksek Lisans Tez, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, 2008.

54. Vapnik, V.N., The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

55. Koç San, Dilek; Mustafa, T., Destek Vektör Makineleri ile Yüksek Çözünürlüklü Görüntülerden Binaların Belirlenmesi, TMMOB Harita ve Kadastro Mühendisleri Odası Ankara Şb., 2008, 173–186.

56. Cohen, J., A Coefficient of Agreement for Nominal Scales, Educational and Psychological Measurement, (1960) 37–46.

57. Powers, D.M.W., Evaluation: From Precision, Recall and F-measure to Roc, Informedness, Markedness & Correlation, Journal of Machine Learning Technologies, 2,1 (2011) 37–63

6. EKLER

Ek 1. XML Çıktısı

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="5.3.015">
  <context>
    <input/>
    <output/>
    <macros/>
  </context>
  <operator activated="true" class="process" compatibility="5.3.015" expanded="true" name="Process">
    <parameter key="encoding" value="UTF-8"/>
    <process expanded="true">
      <operator activated="true" class="set_macros" compatibility="5.3.015" expanded="true" height="60" name="Set Macros" width="90" x="112" y="30">
        <list key="macros">
          <parameter key="path" value="D:\belgeler\akademik\yuksektez\data\"/>
        </list>
      </operator>
      <operator activated="true" class="read_csv" compatibility="5.3.015" expanded="true" height="60" name="Read Data" width="90" x="112" y="165">
        <parameter key="csv_file" value="%{path}\divan.csv"/>
        <parameter key="first_row_as_names" value="false"/>
        <list key="annotations">
          <parameter key="0" value="Name"/>
        </list>
        <parameter key="locale" value="Turkish (Turkey)"/>
        <parameter key="encoding" value="UTF-8"/>
        <list key="data_set_meta_data_information">
          <parameter key="0" value="yazar.true.text.label"/>
          <parameter key="1" value="eser.true.text.attribute"/>
        </list>
      </operator>
      <operator activated="true" class="text:process_document_from_data" compatibility="5.3.002" expanded="true" height="76" name="Process Documents from Data" width="90" x="313" y="165">
        <parameter key="keep_text" value="true"/>
        <parameter key="prune_method" value="absolute"/>
        <parameter key="prune_above_percent" value="100.0"/>
        <parameter key="prune_below_absolute" value="10"/>
        <parameter key="prune_above_absolute" value="10000"/>
        <list key="specify_weights"/>
        <process expanded="true">
          <operator activated="true" class="text:replace_tokens" compatibility="5.3.002" expanded="true" height="60" name="Replace Tokens (3)" width="90" x="112" y="30">
```

Ek 1. XML Çıktısı'nın devamı

```
<list key="replace_dictionary">
          <parameter key="I" value="ı"/>
          <parameter key="İ" value="i"/>
         </list>
        </operator>
        <operator activated="true" class="text:transform_cases" compatibility="5.3.002" expanded="true" height="60" name="Transform Cases (3)" width="90" x="313" y="30"/>
        <operator activated="true" class="text:tokenize" compatibility="5.3.002" expanded="true" height="60" name="Tokenize" width="90" x="514" y="30"/>
        <operator activated="true" class="text:filter_by_length" compatibility="5.3.002" expanded="true" height="60" name="Filter Tokens (by Length)" width="90" x="715" y="30">
         <parameter key="min_chars" value="3"/>
         <parameter key="max_chars" value="10000"/>
        </operator>
        <operator activated="true" class="text:generate_n_grams_terms" compatibility="5.3.002" expanded="true" height="60" name="Generate n-Grams (Terms)" width="90" x="916" y="30"/>
        <connect from_port="document" to_op="Replace Tokens (3)" to_port="document"/>
        <connect from_op="Replace Tokens (3)" from_port="document" to_op="Transform Cases (3)" to_port="document"/>
        <connect from_op="Transform Cases (3)" from_port="document" to_op="Tokenize" to_port="document"/>
        <connect from_op="Tokenize" from_port="document" to_op="Filter Tokens (by Length)" to_port="document"/>
        <connect from_op="Filter Tokens (by Length)" from_port="document" to_op="Generate n-Grams (Terms)" to_port="document"/>
        <connect from_op="Generate n-Grams (Terms)" from_port="document" to_port="document 1"/>
        <portSpacing port="source_document" spacing="0"/>
        <portSpacing port="sink_document 1" spacing="0"/>
        <portSpacing port="sink_document 2" spacing="0"/>
       </process>
      </operator>
      <operator activated="true" class="store" compatibility="5.3.015" expanded="true" height="60" name="Store Word List" width="90" x="514" y="300">
       <parameter key="repository_entry" value="Word_list"/>
      </operator>
      <operator activated="true" class="x_validation" compatibility="5.3.015" expanded="true" height="112" name="Validation" width="90" x="514" y="120">
       <process expanded="true">
        <operator activated="true" class="polynomial_by_binomial_classification" compatibility="5.3.015" expanded="true" height="76" name="Polynominal by Binominal Classification" width="90" x="179" y="120">
         <process expanded="true">
```

Ek 1. XML Çıktısı'nın devamı

<operator activated="true" class="support_vector_machine" compatibility="5.3.015" expanded="true" height="112" name="SVM" width="90" x="447" y="165"/>
<connect from_port="training set" to_op="SVM" to_port="training set"/>
<connect from_op="SVM" from_port="model" to_port="model"/>
<portSpacing port="source_training set" spacing="0"/>
<portSpacing port="sink_model" spacing="0"/>
</process>
</operator>
<connect from_port="training" to_op="Polynominal by Binominal Classification" to_port="training set"/>
<connect from_op="Polynominal by Binominal Classification" from_port="model" to_port="model"/>
<portSpacing port="source_training" spacing="0"/>
<portSpacing port="sink_model" spacing="0"/>
<portSpacing port="sink_through 1" spacing="0"/>
</process>
<process expanded="true">
<operator activated="true" class="apply_model" compatibility="5.3.015" expanded="true" height="76" name="Apply Model" width="90" x="45" y="30">
<list key="application_parameters"/>
<parameter key="create_view" value="true"/>
</operator>
<operator activated="true" class="performance_classification" compatibility="5.3.015" expanded="true" height="76" name="Performance" width="90" x="180" y="30">
<parameter key="kappa" value="true"/>
<parameter key="weighted_mean_recall" value="true"/>
<parameter key="weighted_mean_precision" value="true"/>
<parameter key="correlation" value="true"/>
<list key="class_weights"/>
</operator>
<connect from_port="model" to_op="Apply Model" to_port="model"/>
<connect from_port="test set" to_op="Apply Model" to_port="unlabelled data"/>
<connect from_op="Apply Model" from_port="labelled data" to_op="Performance" to_port="labelled data"/>
<connect from_op="Performance" from_port="performance" to_port="averagable 1"/>
<portSpacing port="source_model" spacing="0"/>
<portSpacing port="source_test set" spacing="0"/>
<portSpacing port="source_through 1" spacing="0"/>
<portSpacing port="sink_averagable 1" spacing="0"/>
<portSpacing port="sink_averagable 2" spacing="0"/>
</process>
</operator>
<operator activated="true" class="multiply" compatibility="5.3.015" expanded="true" height="94" name="Multiply" width="90" x="715" y="210"/>

Ek 1. XML Çıktısı'nın devamı

```
<operator activated="true" class="store" compatibility="5.3.015" expanded="true" height="60" name="Store Performance" width="90" x="849" y="345">
      <parameter key="repository_entry" value="Performance"/>
    </operator>
    <operator activated="true" class="store" compatibility="5.3.015" expanded="true" height="60" name="Store Model" width="90" x="648" y="30">
      <parameter key="repository_entry" value="Model"/>
    </operator>
    <connect from_op="Read Data" from_port="output" to_op="Process Documents from Data" to_port="example set"/>
    <connect from_op="Process Documents from Data" from_port="example set" to_op="Validation" to_port="training"/>
    <connect from_op="Process Documents from Data" from_port="word list" to_op="Store Word List" to_port="input"/>
    <connect from_op="Validation" from_port="model" to_op="Store Model" to_port="input"/>
    <connect from_op="Validation" from_port="training" to_port="result 1"/>
    <connect from_op="Validation" from_port="averagable 1" to_op="Multiply" to_port="input"/>
    <connect from_op="Multiply" from_port="output 1" to_port="result 2"/>
    <connect from_op="Multiply" from_port="output 2" to_op="Store Performance" to_port="input"/>
    <portSpacing port="source_input 1" spacing="0"/>
    <portSpacing port="sink_result 1" spacing="0"/>
    <portSpacing port="sink_result 2" spacing="0"/>
    <portSpacing port="sink_result 3" spacing="0"/>
  </process>
 </operator>
</process>
```

ÖZGEÇMİŞ

1986 yılında Erzurum'da doğdum. İlkokulu Erzurum ve Trabzon'da okuduktan sonra ortaokul ve liseyi Trabzon Kanuni Anadolu Lisesinde kesintisiz bir şekilde okuyarak 2003 yılında mezun oldum. Yine aynı yıl başladığım Başkent Üniversitesi Fen Edebiyat Fakültesi İstatistik ve Bilgisayar Bilimleri Bölümündeki lisans eğitimimi %100 burslu olarak 2008 yılında tamamladım. 2009 yılı Mayıs ayında Iğdır İl Jandarma Komutanlığı'nda zorunlu askerlik görevimi yerine getirdim. 2009 yılında ikinci üniversite olarak başladığım Anadolu Üniversitesi, Açıköğretim Fakültesi, İktisat Bölümünden 2015 yılında mezun oldum. Aynı süreçte 2012 yılında Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstatistik Bölümü Yüksek Lisans Programına kabul edildim ve halen burada öğrenim hayatıma devam etmekteyim. 2010 yılı Kasım ayında TÜİK Uzman Yardımcısı olarak başladığım görevimi, Türkiye İstatistik Kurumu Trabzon Bölge Müdürlüğü'nde halen TÜİK Uzmanı olarak sürdürmekteyim.