YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ TEKNİKLERİ İLE METİNLERİN OTOMATİK OLARAK SINIFLANDIRILMASI

Matematik Müh. Aysun DOĞRUSÖZ

FBE Matematik Müh. Anabilim Dalında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Yrd. Doç. Dr. Nilgün GÜLER BAYAZIT

İSTANBUL, 2007

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

c_i	Sınıflamada kullanılacak verilerin ait olduğu sınıflar
$C_{ÇNB}$	ÇNB sınıflayıcı fonksiyonu
C_{TNB}	TNB sınıflayıcı fonksiyonu
d_i	Sınıflandırılacak dokümanların ifadesi
D	Eğitim ve test dokümanlarının bulunduğu tanım kümesi
ξ	İndirgeme çarpanı
$freg(d,t)$	t. kelimenin d. dökümanda geçme adedini tutan fonksiyon
$f(TDF)$	Terim değerlendirme fonksiyonu
$IDF(t)$	Ters doküman sıklığı fonksiyonu
K	K-EYK sınıflandırıcısının parametresi
R^t	t boyutlu özellik uzayının boyutu
T	Orijinal özellik uzayının boyutu
T'	İndirgenmiş özellik uzayının boyutu
$TF(d,t)$	t. kelimenin d. dokümanla ilişkisini belirten fonksiyon
t_i	Dokümanları oluşturan kelime indeksi
$O\|D\|$	Sınıflama algoritması karmaşıklığı
α	Düzeltme parametresi
θ_{ci}	ÇNB ve TNB sınıflandırıcılarına ait sınıflama parametresi
w_{ci}	ÇNB ve TNB sınıflandırıcılarına ait ağırlık değeri

KISALTMA LİSTESİ

TNB	Tamamlayıcı Naive Bayes
OK	Oransal Kazanç
BK	Bilgi Kazancı
K-EYK	K En Yakın Komşuluk
ÇNB	Çoklu Naive Bayes
MS	Metin Sınıflama
NB	Naive Bayes
TDF	Terim Değerlendirme Fonksiyonu
10'lu ÇG	10'lu Çapraz Geçerleme

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

ÖNSÖZ

Metin sınıflama, kategorileri daha önceden belirlenmiş olan deneme dokümanlarının eğitilmesi ile kategorisi bilinmeyen bir test dokümanının sınıflandırılması problemidir. Günümüzün popüler konularından olan bu problemdeki en önemli sorunlar, sınıflama fonksiyonunun oluşturulması ve özellik uzayı olarak bilinen kümenin büyük bir boyuta sahip olmasıdır.Belirtilen sorunların giderilmesi tam bir mühendislik problemidir.Tez çalışmasında bu sorunlar incelenmiş ve metin sınıflamada kullanılan sınıflama algoritmaları ile özellik seçim teknikleri ele alınmıştır.Böyle bir çalışmanın gerçek hayatta pratiklik sağlaması ve geniş alanlarda kullanılması, gerek çalışma süresince gerekse sonuç aşamasında oldukça faydalı ve zevkli bir süreç geçirmeme vesile olmuştur.

Bu zevkli çalışmamda analiz, modelleme ve uygulama kısmında benden desteğini esirgemeyen tez danışmanım saygıdeğer hocam Yrd. Doç. Dr. Nilgün GÜLER BAYAZIT'a, Doç. Dr. Selim AKYOKUŞ'a ve yüksek lisans süresince çalışmalarımı rahat bir şekilde sürdürmemi sağlayan TÜBİTAK kurumuna teşekkürlerimi sunarım.

Saygılarımla,

Aysun DOĞRUSÖZ

ÖZET

Bu tezin amacı, veri madenciliğinin bir alt dalı olan metin madenciliği kapsamında günümüzün en önemli sorunlarından olan metin sınıflama algoritmalarının matematiksel modelinin incelenmesi ve bir uygulamasının yapılmasıdır. Literatürde bu konu ile ilgili birçok çalışma yapılmıştır. Tez çalışmasında bu çalışmalar incelenmiş ve sonuçlar yorumlanmıştır.

Tez çerçevesinde ilk önce metin madenciliği ve metin sınıflama problemi tanıtılacaktır. Ardından metin sınıflama probleminin matematiksel ifadesi üzerinde durulacaktır. Metin sınıflama problemi uygulanmadan önce problemde kullanılacak olan metinsel verilerin üzerinde yapılan ön işlem aşamaları incelenecektir. Ön işlem aşamalarından sonra metinlerden bilgi çıkarımı, vektör uzay modeli ve metinlerin doğru sınıflandırılma yüzdesinin arttırılmasına yönelik geliştirilen özellik seçim algoritmaları ele alınacaktır. Bunun yanı sıra metin sınıflama için geliştirilmiş olan sınıflama algoritmalarından bahsedilecek, bu algoritmaların olumlu ve olumsuz yönleri üzerinde durulacak ve algoritmalar kıyaslanacaktır.

Tezin son bölümünde ise algoritma performanslarının sayısal sonuçları yer alacaktır. Elde edilen sonuçlardan yola çıkarak özellik seçim algoritmalarının sınıflama yüzdesini ne ölçüde etkilediği tartışılacaktır. Sonuçlar tablolar yardımı ile yorumlanacak ve öneriler sunulacaktır.

Anahtar Kelimeler: Metin Sınıflama, Vektör Uzay Modeli, Naive Bayes, Çoklu Naive Bayes, Tamamlayıcı Naive Bayes, En Yakın K Komşuluk, Bilgi Kazancı, Oransal Kazanç, Ki-Kare.

ABSTRACT

The aim of this thesis is to study the mathematical model of text classification algorithms, being one of the most important problems of present time, and then, to apply it to the real life by using the context of text mining which is a sub branch of data mining. There have been a lot of studies in literature about this subject. In this thesis, these previous workings will be studied while their results will be interpreted.

In the frame of this thesis, firstly, text mining and the problem of text classification will be introduced. After that, the formulation of classification problem will be explained. Before applying the text classification problem, the pre-calculation processes of text data will be clarified to use in the problem. Following these pre-calculation processes; being aimed at vector-space model, at taking information from text and at increasing the percent of correct classifying, the feature selection algorithms will be discussed. Besides, developed for text classification, the classification algorithms will be illustrated and then compared with each other by presenting their positive and negative sides.

In the last section, there are the numerical results of algorithm performances. The influence of feature selection algorithms on the classification percent will be argued by using obtained results. Final results will be interpreted with the help of graphs and presented in tableaus.

Keywords: Text Classification, Vector-Space Model, Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, K-Nearest-Neighbor, Information Gain, Gain Ratio, Chi-Square

1. GİRİŞ

1.1 Konu ve Önemi

Teknolojinin ilerlemesi ile elektronik formda tutulan metinsel verilerin sayısı günden güne artmaktadır. Kurumlar, verilerinin büyük bir kısmını metinsel olarak saklamaktadır. Kurumlar açısından bu verilerin analiz edilmesi ve gerekli verilere kısa sürede erişilmesi büyük önem taşımaktadır. Metinsel verilerin analiz edilmesinin önem kazanması metin sınıflama problemini günümüzün popüler konuları arasına koymuştur.

Metin Sınıflama (MS), makine öğrenmesi teknikleri ile metinlerin otomatik olarak sınıflandırılması eylemidir. Amaç, dokümanların nasıl sınıflanması gerektiğini tanımlayan sınıflayıcı bir fonksiyon oluşturmaktır. Sınıflayıcının yapılandırılması için geliştirilmiş birçok algoritma mevcuttur. Bu algoritmalardan **Naive Bayes (NB)**'i temel alan sınıflama fonksiyonları, bazı araştırmacılar (Domingos ve Pazzani, 1997; Lewis, 1992, 1998; McCallum, 1998; Dumais, Heckerman ve Sahami, 1998) tarafından geniş biçimde incelenmiştir. Onların NB sınıflamalarında bir doküman, her kelimesinin varlığını veya yokluğunu temsil eden ikili bir düzen vektörü olarak düşünülür. Bu NB'e, çoklu rastsal değişkenli Bernoulli Naive Bayes denir. Ancak bu model, dokümandaki terim frekanslarından faydalanmak için uygun değildir. Bundan dolayı, **Çoklu Nominal Naive Bayes (ÇNB)**, Naive Bayes metin sınıflamacılarına bir alternatif olarak öne sürülür. Son yıllarda birçok araştırmacı standart NB metin sınıflamacıları (Yang ve Lui, 1999; McCallum, 2000; Kim, Han, Rim ve Myaeng, 2005) olarak buna göz atmışlardır. ÇNB algoritmasının eksik yönlerinin giderilmesi amacı ile **Tamamlayıcı Naive Bayes (TNB)** algoritması ortaya çıkmıştır(Rennie, Shih, Teevan ve Karger, 2003). Bu algoritmada ÇNB'de eksik yönlerin giderilmesine yönelik çalışmalar, sınıflama fonksiyonu parametrelerinin değişik seçimlerle yapılması sonucu ortaya çıkmıştır.

Bazı metin sınıflamacıları, **K-En Yakın Komşuluk (K-EYK)** algoritmasını temel almıştır(Soucy ve Mineau, 2001). Bu araştırmacılar, çeşitli ağırlık değerlerine göre dokümanları temsil eden vektörleri oluşturup benzer dokümanların benzer vektörlere sahip olduğu görüşünü anlatmaya çalışmışlardır.

Sınıflama probleminde sınıflayıcı algoritmaların yanında özellik seçim teknikleri de incelenmektedir. Metinsel verilerin büyüklüğü sınıflama algoritmalarının verimliliğini olumsuz etkilemektedir. Bu amaçla yapılan çalışmalarda (Shen ve Jiang, 2003; Hall ve

Holmes, 2003; Shang, Huang, Zhu, Lin, Qu ve Wang, 2007) sınıflama için önem derecesi yüksek olan kelimeler seçilmiştir. Kelime seçimi için kullanılan matematiksel ifadeler çalışmalara göre değişkenlik göstermektedir. Tez çalışmasında kullanılan özellik seçim tekniklerinden **Bilgi Kazancı(BK)**, **Oransal Kazanç(OK)** ve **Ki-Kare** algoritmaları incelenmiştir.

Kısacası tez çalışmasında metin sınıflama problemi, bahsedilen özellik seçim ve sınıflama algoritmaları ile incelenecek ve belirlenen bir veri kümesi üzerinde bu tekniklerin uygulaması gerçekleştirilecektir.

1.2 Tezin Amacı ve Kapsamı

Elektronik formda erişilebilir halde bulunan çok sayıda doküman mevcuttur ve sayıları teknoloji ile birlikte her geçen gün artmaktadır. Web, tek başına milyarlara yaklaşan sayıda doküman içermektedir. Bu kadar fazla sayıda bulunan bilgi yığınlarına ulaşmak için düzenli bir sistemin oluşturulması şarttır.

Günümüzde, çoğu web sitesi sahibi sitelerinin hiyerarşik bir organizasyona göre yapılandırılmasını, elektronik posta kullanıcıları almış oldukları iletilerin filtrelenmesini, akademik komiteler sitelerde yayınlanacak olan makale ya da diğer yayınlarının belli bir düzende olmasını beklemektedir. Bu düzenlemeleri gerçekleştirmek için kurallar oluşturmak ve kuralları uygulanabilir kılmak için sınıflamaları elle gerçekleştirmeye çalışmak yoğun çaba gerektirir. Bu nedenle otomatik sınıflandırma sistemleri oluşturulmuştur. Bu sistemler kullanılarak belirtilen işlemler kolaylıkla uygulanabilmektedir. İşte tez çalışmasının amacı, bu sınıflama sistemlerini incelemektir.

Metin sınıflamada izlenen temel aşamalar; metinlerden bilgi çıkarmak, çıkarılan bilgileri analiz etmek ve analiz sonuçlarına bağlı olarak metinleri kategorize etmektir. Tez çalışmasında Basit Naive Bayes, Çoklu Naive Bayes(Multinomial Naive Bayes), Tamamlayıcı Naive Bayes(Complement Naive Bayes) ve K-En Yakın Komşuluk(K-Nearest Neighbor) sınıflama algoritmaları ile birlikte Bilgi Kazancı(Information Gain), Oransal Kazanç(Gain Ratio) ve Ki-Kare(Chi-Square) özellik seçim algoritmaları incelenmiştir. Bu algoritmalar 10 kategoriden oluşan 1000 doküman üzerinde uygulanmış ve sonuçları yorumlanmıştır. Algoritmalar kendi aralarında kıyaslanmış ve algoritmaların geliştirilmesi için öneriler sunulmuştur.

2. METİN MADENCİLİĞİ VE METİN SINIFLAMA PROBLEMİ

Veri madenciliği alanında şu ana kadar yapılan çalışmalar incelendiğinde kullanılan verilerin ilişkisel veri yapılarına dayandıkları görülür. Bu verilerin büyük oranını metin veri tabanları oluşturmaktadır. Metin veritabanları; makaleler, gazeteler, yayınlar, dijital kütüphaneler, elektronik postalar ve web sayfaları gibi çok çeşitli kaynaklardan alınmış dokümanları içermektedir. İnternet üzerindeki dokümanların hızla artması birbirine bağlı metin veri tabanlarının önemini arttırmıştır. Günümüzde idari teşkilatlar, endüstriyel kuruluşlar ve diğer birçok kurum bilgilerini elektronik olarak metin veri tabanlarında saklamaktadır.

Metin veri tabanlarında tutulan dokümanlar yapısal ya da yapısal olmayan dokümanlar olabilir. Bir doküman; başlık, yazar, yayınlanma tarihi, kategori ve bunun gibi konuları içeriyorsa yapısaldır. Yapısal dokümanlardan metnin içeriği ile ilgili bilgilere ulaşılabilir. Eğer bir doküman sadece özet ve konu anlatımından oluşuyorsa yapısal değildir. Kurumların sahip olduğu verilerin %80'den fazlası yapısal olmayan dokümanlardan oluşmaktadır. Bu dokümanlar içerisinde yüzlerce önemli bilgi gizlenmektedir. Kurumlar açısından bu veriler, hizmet sağladıkları insanlarla ilişkilerinin güçlenmesi ve onlara kendi ifadeleri ile görüş bildirmesi açısından önem taşımaktadır. Bu metinlerin kısa sürede analiz edilmesi ve nitelikli bilgilere erişilmesi önemlidir. Yapısal olmayan bu tarz dokümanların sınıflandırılması metin madenciliğinin araştırma konularındandır.

Metin madenciliğinin bu alandaki faydalarını şu şekilde sıralayabiliriz:

- Yapısal olmayan veri kaynaklarındaki değeri ortaya çıkarır.
- Doküman kaynakları yönetiminin otomatikleştirilmesini sağlar.
- Veri madenciliği ve metin madenciliği birlikte kullanılarak kurumsal başarı arttırılır.
- Hizmet ettikleri insanları, kendi ifadeleriyle analiz ederek kuruma yeni bir anlama yolu sağlar.
- İstihbarat teşkilatları yapısal verileri ve metin halindeki verileri (telefon kayıtları, web sayfa içerikleri, toplanan her türlü yazılı bilgi) gerçek anlamda değerlendirebilirler.

Metin veri tabanları ihtiyacının artmasıyla birlikte geleneksel bilgi erişim teknikleri yetersiz kalmaya başlamıştır. Metinlerin içeriklerini tam olarak bilmeden analiz edebilmek için efektif sorgular yapmak ve gerekli olan bilgilere ulaşmak zordur. Bu nedenle kullanıcılar farklı dokümanları kıyaslayan ve dokümanlar arası ilişkiyi belirginleştirici faktörleri ortaya çıkaran algoritmalar geliştirmeye çalışmışlardır. Metinleri kategorize etmede geliştirilen çalışmaların

bazılarında, metinleri oluşturan kelimelerin geçme sıklığına bağlı olarak istatistiksel olasılık değerleri hesaplanmıştır. Bazı durumlarda ise istatistiksel metotlar yetersiz kalmış, kelimelerin anlamlarına ve metnin gramer yapısına bakılmıştır. Uygulanan farklı metotlarla elde edilen bilgileri modellemek ve elverişli algoritmalar oluşturmak için çok geniş bilgi erişim çalışmaları gerçekleştirilmiştir. Tez çalışmasında bu alandaki gelişmeler de incelenmiştir.

2.1 Metin Sınıflama Problemi

Metin sınıflama, dokümanlardan oluşan bir veritabanında dokümanların konularına göre sınıflandırılmasıdır. Amaç, metin madenciliği teknikleriyle yazılı belgeler arasındaki ilişkileri incelemek ve örüntüleri bulmaktır.

Metin sınıflamada izlenmesi gereken aşamalar şu şekilde belirtilebilir:

Kullanılacak olan verilerin tespit edilmesi,

⇩

Metinlerin ön işlem aşamalarından geçirilmesi,

⇩

Sınıflandırma için verilerden bilgi çıkarılması,

⇩

Özellik uzayının boyutunun azaltılması adına

sınıflama için önemli kelimelerin

belirlenmesi

⇩

Metinlerin sınıflandırılması.

Sınıflama probleminde ilk aşama veri seçimi aşamasıdır. Sınıflamada kullanılacak olan veriler metinsel olmalıdır ve metinlerin bulunmasında bazı hususlara dikkat edilmelidir:

- Doğal dilde yazılmış metinler bulunmalıdır.
- Birbiriyle ilişkili belgeler bulunmalıdır.
- Aynı konudaki belgeler bulunmalı ve gruplandırılmalıdır.
- Bulunan belgeler sıralanmalıdır.

İkinci aşama, dokümanların ön işleme tabi tutulduğu aşamadır. Uygulanan ön işlemler aşağıda sıralanmıştır:

- Metin içindeki terimleri ayıklama,
- HTML etiketlerinden arındırma,
- Farklı terimleri belirleme,
- Kök bulma (Aynı kökten gelen farklı ek almış sözcüklerin köklerini bulma),
- Sık geçen sözcükleri ayıklama (bağlaçlar, edatlar, sık kullanılan filler vs...).

2.2 Problemin Formülasyonu

Metin sınıflama ile önceden tanımlanmış bir $C= \{c_1, ... , c_{|c|} \}$ kümesinin tanım kümesi olan D kümesi altında, doğal dil metinlerinin kategorize edilmesi incelenmektedir. Daha biçimsel olarak ifade edilirse amaç dokümanların nasıl sınıflanması gerektiğini tanımlayan, bilinmeyen bir $\Phi: D \times C \rightarrow \{0,1\}$ hedef fonksiyonunu, sınıflayıcı olarak tabir edilen $\Phi^{\wedge}: D \times C \rightarrow \{0,1\}$ fonksiyonuna mümkün olduğu kadar yaklaştırmaktır.

Bir metnin sınıflayıcısının yapılandırılması, C kümesi altında önceden sınıflanmış dokümanların bir ana başlangıç bölgesi olan $\Omega=\{d_1, ... ,d_{|\Omega|}\}$'in varlığına bağlıdır. Eğiticili öğrenme olarak tabir edilen genel bir tümevarımsal süreç $T_r=\{d_1, ... ,d_{|Tr|}\}$ deneme kümesinden C kümesinin özelliklerini öğrenen bir sınıflayıcı inşa eder. Sınıflayıcı inşa edildiğinde de onu $Te=\Omega-Tr$ şeklindeki test kümesinde uygulayarak ve sınıflayıcının kararları ile ana bölgede kodlananlar arasındaki uygunluk derecesini kontrol ederek, sınıflayıcının etkinliği test edilir. Öğrenme, kategorilerdeki deneme dokümanlarının eğitilmesi ile gerçekleştiğinden eğiticili öğrenme adını alır. Bir metin sınıflayıcının inşası doküman derecelendirme ve doküman sınıflama safhalarından oluşur. Bu safhalar aşağıda açıklanmıştır:

1) Doküman derecelendirme safhası, dokümanlar için iç temsillerin yaratılma aşamasıdır. Bu da tipik olarak şunları içerir:

a) Terim seçme safhası, özellik uzayını simgeleyen T kümesini en etkin şekilde temsil edecek veya verimlilik ile etkinlik arasındaki en iyi uyumu verecek olan $T' \subset T$ alt kümesinin seçimini içermektedir. Başka bir ifadeyle bu safha özellik uzayının boyutsallık indirgemesinin bir biçimidir.

b) Terim ağırlıklandırma safhası, özellik uzayını oluşturan bütün t_k terimleri (1a) safhasındaki gibi seçildikten sonra, t_k teriminin kaç adet dj dokümanında bulunduğunu gösteren $0 \leq w_{kj} \leq 1$ şeklinde ağırlık değerleri hesabını içerir.

Terim ağırlıklarının hesaplanması ele alındığında, (1b) safhasının (1a)'nın sonuçlarından faydalandığı görülmektedir. Çünkü en iyi terimlerin seçimi, genelde c_i kategorisini ayırt etme kapasitesini ölçen bir $f(t_k,c_i)$ terim seçme fonksiyonun oluşturulması ve daha sonra da bu fonksiyonu maksimum kılan terimlerin seçilmesi vasıtasıyla her t_k teriminin değerlendirilmesiyle başarılır.

2) Doküman sınıflama safhası, deneme dokümanlarının iç temsillerinden öğrenilen bir sınıflayıcının yaratılması sürecini içerir. Problemin en önemli kısmıdır.

3. BİLGİ ERİŞİM SİSTEMLERİ

Teknolojinin ilerlemesiyle birlikte günümüzde veritabanlarında büyük miktarlarda sınıflandırılmamış metinsel bilgiler tutulmaktadır. Bu bilgilerden ilginç ve önemsiz olmayan örüntüleri bulmak, bilgi çıkarımı için düzensiz ve sayıca fazla olan dokümanları analiz etmek tipik bir bilgi çıkarım sistemi problemidir. Kullanıcı sorgusuna dayalı olan bir doküman koleksiyonuna dışarıdan gelen bir dokümanı ilgili yere yerleştirebilmek için karakteristik anahtar kelimeler verme yöntemi kullanılır. Dokümanlara erişmek bu anahtar kelimelerle gerçekleştirilir. İkinci el araba almak isteyen birinin araştırması ile ilgili gerekli bilgiye ulaşması için gereken anahtar sözcük arabanın markası olabilir. Eğer bu kısa bilgi kullanıcı için yetersizse sorgulama için daha uzun bilgilere ihtiyaç duyulur. Bu tarz bilgi giriş işlemleri bilgi filtreleme olarak adlandırılmaktadır. Metinlere ulaşma amacı ile iç temsilciler yaratma çalışmaları bilgi erişim sistemlerini meydana getirmiştir. Çıkarım bilgilerinden sonra doğru dokümana hangi oranda ulaşıldığı doğruluk oranının bulunması ile tespit edilir. Bu oran çeşitli hesaplamalarla gerçekleştirilmektedir ve bu şekilde algoritmaların doğru sınıflama yüzdeleri hesaplanmaktadır.

Bilgi çıkarımı (Information Retrieval) veri tabanı sistemleriyle paralel olarak çalışan bir alandır. Veri tabanı sistemleri, sorgulama ve yapısal olan verilerin transferiyle ilgilenir. Bilgi erişim sistemleri ise yapısal olmayan metin tabanlı dokümanlardan bilgi çıkarımı ve bilgi organizasyonu ile ilgilenmektedir. Veri tabanı ve bilgi erişim sistemlerinin ilgilendikleri verilerin farklı olması inceledikleri problemlerinde farklı olmasına sebep olmuştur. Veri tabanı sistemleri eş zaman kontrolü, iyileştirme, transfer yönetimi ve güncelleme konularını işlemektedir. Bilgi erişim sistemleri ise yapısal olmayan dokümanları ve bu verilerden bilgi çıkarımı konularını işler.

Metinsel verilerden bilgi çıkartmak için bilgi erişim sistemlerince ortaya konulmuş olan yöntemler kullanılmaktadır. Genellikle kullanılan iki metot vardır. Bunlar:

a) Doküman Seçme Metodu (Document Selection Method)

b) Doküman Derecelendirme Metodu (Document Ranking Method)

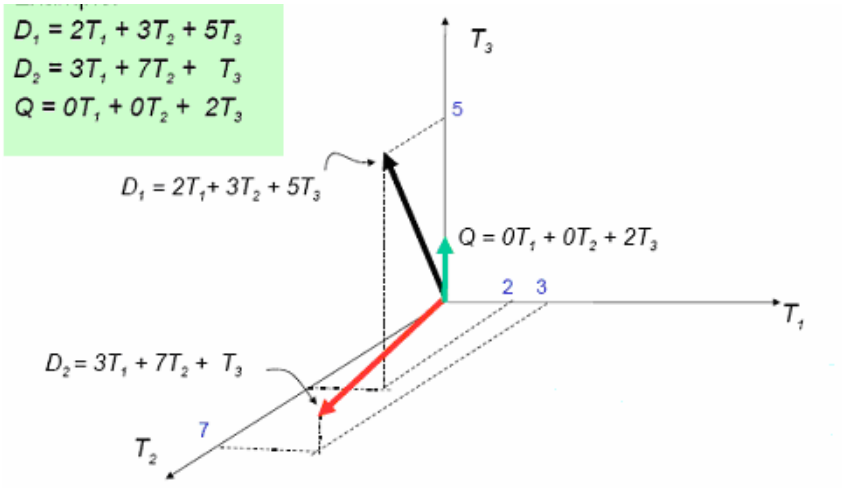
Doküman seçme metodunda sorgular ilgili dokümanları bulmak için dokümanlara özel anahtar sözcükler verilerek yapılır. Bu kategoride kullanılan klasik metot Boolean Bilgi Çıkarım Metodudur (Boolean Retrieval Model). Bu modelde, bir doküman kullanıcı tarafından tanımlanmış olan anahtar sözcüklerle temsil edilir. Boolean modelinde kullanılan anahtar sözcük ifadeleri “car **and** repair shop”, “ tea **or** coffee”, “database systems **but not**

oracle" şeklindedir. Bu şekilde tanımlanmış Boolean sorguları bu sorgularla ilgili dokümanları geri döndürür. Ancak dokümanları bu şekilde düzenlemek ancak kullanıcı dokümanla ilgili birden fazla veriye sahipse efektif olur.

Doküman derecelendirme metodunda ise dokümanlar ilgili olma durumlarına göre derecelendirilir. Genel kullanıcılar için bu metot doküman seçme metodundan daha uygundur. Çeşitli matematiksel ifadeleri (lojik, olasılık, istatistik) kabul eden çok sayıda derecelendirme yöntemi vardır. Tez çalışmasında bu yöntemler ele alınacaktır.

3.1 Vektör Uzay Modeli

Vektör uzay modeli; bilgi çıkarımı, bilgi filtreleme ve veri indeksleme gibi alanlarda kullanılan cebirsel bir modeldir. Doğal dil belgelerinin çok boyutlu uzayda özel bir anlamını simgelemektedir. Vektör uzay modeli, metin veritabanını oluşturan dokümanlar ve bu dokümanlara ait kelimeleri yüksek boyutlu uzayda vektör olarak tutmaktadır. Çeşitli hesaplamalarla her bir dokümana ait kelimeler için ağırlık değerleri tespit edilir ve hesaplanan bu ağırlık değerlerine göre dokümanların benzerlik ilişkileri ortaya konulur.



Şekil 3-1 Vektör uzay modelinin 3 boyutlu koordinat sisteminde gösterimi.

Vektör uzayı modelinin uygulanması için gereken ilk adım dokümanları temsil edecek olan kelimelerin belirlenmesidir. Bu kelimeler, dokümanlardaki sözcüklerin birbirinden bağımsız olacak şekilde ayrılmasıyla belirlenir. Yani bu işlem için dokümanlar atomlarına ayrılmalıdır. Gereksiz olan kelimelerin indekslenmesine engel olmak için İngilizcede çok sık kullanılan ve

sınıflandırma için ayırt edici olmayan kelimeler her dokümandan ayrılmalıdır. Bu kelimelere sık kullanılan gereksiz sözcükler (stopwords) denilmektedir. Gereksiz kelimeler doküman sınıflandırmada etkili olmayan, sık kullanılan *a, the, of, for, with, and, so on,* gibi zamir bağlaç, zarf ya da haftanın günleri, yılın ayları ve çok sık kullanılan fiilleri içermektedir. Bu kelimelere internet ortamından erişilebilmektedir.

Dokümanların gereksiz kelimelerden arındırılması vektör uzay modelini oluşturan özellik uzayının boyutunu azaltmaktadır. Bu durum dokümanları sınıflandırmak için gerekli olan sınıflandırma algoritmalarının daha efektif sonuçlar vermesini ve sonuca daha kısa sürede ulaşılmasını sağlayacaktır.

Özellik uzayının boyutunu azaltmak için izlenmesi gereken ikinci adım kelimelerin köklerine ayrılmasıdır. Aynı köke sahip olan kelimeler dokümanlarda farklı ekler alarak bulunabilirler. Kelimeler köklerine ayrılmazsa aynı köke sahip olan kelimeler farklı kelimelermiş gibi algılanır. Bu durum özellik uzayının boyutunu arttırır. Ayrıca uygulanacak olan, gerek özellik seçim algoritmalarının gerekse sınıflandırma algoritmalarının sonucunu olumsuz etkiler. Tablo 3.1'de aynı köke sahip olan farklı ekler almış kelimeler görülmektedir. Uygulanan kök ayırma algoritması ile kelimeler köklerine ayrıldığında aslında hepsinin aynı kelimeden türediği görülür. Böylece özellik uzayının boyutu azaltılmış olur.

Tablo 3-1 Kelimeler ve kökleri.

Kelime	**Kökü**
Accepts	accept
acceptable	accept
acceptence	accept
accepted	accept

3.2 Vektör Uzay Modelinin Matematiksel İfadesi

Veri tabanında d adet doküman seti bulunsun ve bu set t adet kelimeden oluşsun. Her doküman t boyutlu uzaydaki (R^t), *v* vektörü ile gösterilsin. Bu şekilde oluşturulan modelleme "vektör uzay" modellemesidir.

Kelime sıklığı (term frequency), t. kelimenin d.dokümanda geçme sayısını tutmak üzere freg(d,t) olarak gösterilsin. Bu bilgi kullanılarak, ağırlıklı kelime sıklığı matrisi elde edilebilir. TF(d,t), t.kelimenin d.dokümanla olan ilişkisi hakkında bilgi vermektedir. d. doküman eğer

t.kelimeyi içermiyorsa TF(d,t)' ye sıfır atanır, tersi durumda ise bir atanır. Hesaplama çok değişik şekillerde gerçekleştirilebilir. **Cornell SMART** sistemine göre kelime sıklığı normalize edilerek aşağıdaki şekilde hesaplanmıştır:

$$TF(d,t) = \begin{cases} 0 & , \text{eğer } freg(d,t) = 0 \\ 1 + \log(1 + \log(freg(d,t))) & , \text{diğer durumlarda} \end{cases} \quad (3.1)$$

Kelime sıklığı bilgisinin yanında hesaplanması gereken bir bilgide IDF(t) (inverse document frequency)'dir. Bu bilgi t. kelimenin önemini belirtmektedir. IDF(t) şu formüle göre hesaplanmaktadır:

$$IDF(t) = \log \frac{1+|d|}{|d_t|} \quad (3.2)$$

(3.2) denkleminde d veri setindeki doküman sayısı, d_t ise t. kelimenin geçtiği doküman sayısıdır. Eğer $d_t << d$ ise t.kelimenin IDF değeri büyüktür.

Vektör Uzayı Modellemesinin tamamlanması için TF ve IDF değerlerinin birlikte kullanıldığı ölçüm aşağıda görüldüğü gibidir:

$$TF\text{-}IDF(d,t) = TF(d,t) \times IDF(t) \quad (3.3)$$

Örnek :

Aşağıdaki çizelgenin her satırı veri tabanındaki dokümanları, her sütunu ise özellik uzayını oluşturan kelimelerinin her bir dokümanda geçme sayısını göstermektedir. Bu çizelgeye kelime sıklığı matrisi de denmektedir. Örnek ile "Cornell Smart" sisteminin bir uygulaması yapılacaktır. Kelime sıklığı bilgileri çıkarılarak, dokümanları temsil edecek bilgilere erişilecektir. Genel anlamda yapılan işlermler vektör uzay modellemesinin gösterimidir.

Tablo 3-2 Kelime sıklığı matrisi.

	t1	t2	t3	t4	t5	t6	t7
d1	0	4	10	8	0	5	0
d2	5	19	7	16	0	0	32
d3	15	0	0	4	9	0	17
d4	22	3	12	0	5	15	0
d5	0	7	0	9	2	4	12

Cornell Smart sistemine göre gereken hesaplamalar:

$TF(d_4, t_6) = 1 + \log(1+\log(15)) = 1.3377$

$IDF(t_6) = \log \frac{1+5}{3} = 0.301$

$TF * IDF(d_4, t_6) = 1.3377 * 0.301 = 0.403$

Bu ağırlık değerlerinin hesaplanmasındaki amaç, veri tabanında bulunan dokümanların benzerliklerinin tespit edilmesidir. Benzer olan dokümanlar benzer ağırlık değerlerine sahiptir.

Benzerlik, dokümanlar arasındaki uzaklık değerleri hesaplanarak bulunur. Uzaklık değerleri "Öklit Uzaklık Ölçüm Formülüne" göre hesaplanabilmektedir.

$$\| d_i - d_j \| = \text{sqrt} \left(\sum (d_i - d_j)^2 \right) \quad (3.4)$$

Birbirine benzeyen dokümanların arasındaki mesafe küçüktür. Yeni gelen bir test dokümanı bu mesafeye göre sınıflandırılmaktadır. Sınıflama algoritmaları, sınıflayıcıları bu mesafeyi temel alarak oluşturmaktadır.

4. KELİME SEÇİMİ VE DOKÜMAN SINIFLAMA ALGORİTMALARI

Metin sınıflama probleminin ana aşaması sınıflama fonksiyonunu oluşturmaktır. Şu ana kadar yapılan çalışmalar incelendiğinde sıklıkla, sınıflandırılacak dokümanlardaki kelime özelliklerinin birbirinden bağımsız olduğu görülmektedir. Son zamanlarda bu algoritmaların yetersizliği nedeniyle kelimelerin birbirlerine olan etkileri ele alınarak kelimelerin sahip oldukları anlamlar, eş anlamlı kelimeler, kelime türleri tespit edilmiş ve sınıflama fonksiyonları bu özellikler katılarak oluşturulmuştur. Yapılan bazı çalışmalarda, yine kelimeler birbirinden bağımsız olarak kabul edilmiş ancak kullanılan algoritmalar geliştirilmeye çalışılmıştır.

Günümüzde birçok sınıflama algoritması kullanılmaktadır. En çok kullanılan metin sınıflama algoritmaları aşağıda belirtilmiştir:

- K-En Yakın Komşuluk
- Naive Bayes
- Karar Ağacı
- Destek Vektör Makineleri
- Yapay Sinir Ağları

Metin sınıflama probleminde diğer aşamalardan biri, sınıflandırılacak olan metinlerin sahip olduğu özellik uzayının boyutunu azaltacak yöntemleri kullanmaktır. Hedefimiz orijinal özellik uzayından sınıflama için ayırt edici özellikleri elde etmek ve özellik uzayının boyutunu azaltmak olmalıdır. Eğer bu durum gerçekleştirilirse sınıflama için kullanılan algoritmalar daha efektif sonuçlar verecektir.

Özellik belirleme yöntemleri, istatistiksel ve makine öğrenmesi metotlarına dayanmaktadır. En çok bilinen özellik seçim metotları aşağıda belirtilmiştir:

- Bilgi Kazancı Özellik Seçim Metodu
- Oransal Kazanç Özellik Seçim Metodu
- Ki – Kare Özellik Seçim Metodu

4.1 Kelime Seçimi ile Boyut Azaltma Yöntemleri

Metin sınıflama probleminde, sınıflama algoritmalarının hesaplanma maliyeti dokümanları temsil eden vektörlerin uzunluğunda bir fonksiyondur. Bu uzunluk, |T| ile ifade edilsin.

Sınıflamanın efektif bir şekilde gerçekleşmesi için boyut olarak |T| uzunluğundan daha kısa vektörlerle çalışabilmek oldukça önemlidir. Yapılan araştırmalarla, özellik uzayının boyutunu azaltmak amacı ile kelime seçim teknikleri ortaya çıkmıştır. Bu teknikler, T kümesinden dokümanın anlamını temsil etmek için en verimli terimlerin bulunduğu T' altkümesini elde etmeyi amaçlamaktadır.

T', (|T'|<<|T|) koşuluyla oluşturulan T kümesinin bir alt kümesi olmak üzere:

$$\xi = \frac{|T| - |T'|}{|T|} \tag{4.1}$$

şeklinde tanımlansın. Bu değere indirgeme çarpanı denmektedir. Amaç, indirgeme çarpanını sınıflandırmayı en verimli hale getirecek şekilde seçmektir.

Terim seçme teknikleri, terim değerlendirme fonksiyonunu (f (TDF) oluşturmak ve daha sonra f'i maksimum kılan terimlerin bir T' kümesinin seçilmesi vasıtasıyla T kümesindeki her terimin değerlendirilmesini içerir. Eğer otomatik sınıflayıcının boyut indirgemesi ile elde edilen T' alt kümesinin dokümanlarını sınıflamada elde ettiği başarı yüzdesi, T kümesinin dokümanlarını sınıflamada elde ettiği başarı yüzdesinden fazla ise terim seçimi beklenen verimi sağlamıştır.

Çoğunlukla bilgi teorisi ve istatistik geleneğinden gelen birçok fonksiyon, metin sınıflandırmada TDF'ler olarak kullanılmıştır. Bunlardan ortaya konan formüller Tablo 4.1'de gösterilmiştir. Bu tablonun üçüncü kolonunda olasılıklar, dokümanların olay uzayı üzerinde yorumlanmaktadır. Örneğin $P(\overline{t_k}, c_i)$, bir x dokümanı için t_k terimlerinin x'de ortaya çıkmama olasılığıdır. TDF fonksiyonlarının çoğu, c_i'nin pozitif ve negatif örneklerinin kümesinde farklı biçimde dağılmış olan sınıflama için yüksek değere sahip terimleri ele geçirmeye çalışır. Nitekim, bu basit prensibin yorumları farklı fonksiyonlara göre değişkenlik gösterebilir.

Tablo 4-1 Özellik seçim algoritmaları ve matematiksel ifadeleri.

Fonksiyon Adı	**Matematiksel Formu**
Ki-kare $X^2(t_k,c_i)$	$\frac{[P(t_k,c_i)P(\overline{t}_k,\overline{c}i)-P(t_k,\overline{c}_i)P(\overline{t}_k,c_i)]^2}{P(t_k)P(\overline{t}_k)P(c_i)P(\overline{c}_i)}$
Bilgi Kazancı ($BK(t_k,c_i)$)	$\sum_{c\in\{c_i,\overline{c}_i\}}\sum_{t\in\{t_k,\overline{t}_k\}} P(t,c)\log_2\frac{P(t,c)}{P(t)P(c)}$
Oransal Kazanç ($OK(t_k,c_i)$)	$\frac{\sum_{c\in\{c_i,\overline{c}_i\}}\sum_{t\in\{t_k,\overline{t}_k\}} P(t,c)\log_2\frac{P(t,c)}{P(t)P(c)}}{\sum_{c\in\{c_i,\overline{c}_i\}} P(c)\log_2 P(c)}$

Örnek :

|C| = 1 olan bir problem düşünelim. Burada önem verilen temel durum elimizdeki dokümanın belirli olan bu kategoriye ait olup olmadığıdır. 1000 adet eğitim dokümanının bulunduğu bir veritabanında 100 adet doküman bu kategoriye ait, geriye kalan 900 doküman ise bu kategori dışında olsun. Özellik uzayına ait 3 kelimenin (t_1,t_2,t_3), Tablo 4.1'de belirtilen özellik seçim yaklaşımlarıyla sahip olduğu derece değerlerini inceleyelim. Bu kelimeler ile ilgili bilgiler aşağıda sıralanmıştır:

t_1 kelimesi C kategorisine ait 100 dokümandan 90'nında, kategori dışındaki 900 dokümanın ise hiç birinde geçmemektedir.

t_2 kelimesi C kategorisine ait 100 dokümandan hiç birinde geçmemekte, kategori dışındaki 900 dokümandan 800'ünde geçmektedir.

t_3 kelimesi C kategorisine ait 100 dokümandan 1'inde, kategori dışındaki 900 dokümandan 9'unda geçmektedir.

Bu bilgiler altında Tablo 4.1'de hesaplanan özellik seçim formülleri sonucunda elde edilen kelime dereceleri Tablo 4.2'de görülmektedir.

Tablo 4-2 Kelimelerin geçme sıklığına göre hesaplanan derecelendirme değerleri.

	#(t,c)	#(t, $\bar{c}$)	BK(t,c)	X^2(t,c)	OK(t,c)
t_1	090	000	0.267	890.110	0.822
t_2	000	800	0.232	444.444	0.714
t_3	001	009	0.000	000.000	0.000

Özellik uzay vektörünün kelimeleri bu şekilde derecelendirildikten sonra, bir derece sınırı belirlenerek (ϑ), derecesi ϑ'dan büyük olan kelimelerin yeni özellik uzayını oluşturması sağlanmaktadır. Böylece önem derecesi fazla olan kelimeler seçilmiş olur.

4.2 Naive Bayes Sınıflama Algoritması

Naive Bayes (NB), uzun yıllardır kullanılan popüler makine öğrenmesi metotlarındandır. Naive Bayes Teoreminin temel düşüncesi, bir dokümanı birbirinden bağımsız kelimelerin oluşturduğu bir set olarak ele alıp, bu kelimelerin dokümanlarda geçme olasılığına bağlı olarak sınıflama yapmaktır. Teorem aşağıdaki şekilde ifade edilebilir:

$$P(c_j \setminus d_i) = \frac{P(c_j)\, P(d_i \setminus c_j)}{P(d_i)} \quad \text{(posterior probability)} \tag{4.2}$$

Bu denklem, d_i dokümanının c_j kategorisine aitlik yüzdesini vermektedir. Veri tabanındaki d_i dokümanının sınıflandırılması $P(c_j \setminus d_i)$ değerine göre değişmektedir. Algoritmanın uygulanması sonucu en yüksek $P(c_j \setminus d_i)$ değerine sahip olan d_i dokümanı, c_j sınıfına ait olur. Bu ifade matematiksel olarak denklem (4.3)'de belirtilmiştir.

$$P(c_j \setminus d_i) = \arg\max \frac{P(cj)\ P(di \setminus cj)}{P(di)} \ , cj \in C \tag{4.3}$$

Denklem (4.3)'de tüm kategoriler için $P(d_i)$ aynı olacağından, sınıflandırmanın yapılabilmesi

için aşağıdaki hesaplamanın yapılması yeterli olacaktır:

$$NB = \arg\max P(c_j) P(d_i \setminus c_j) \quad (4.4)$$

$P(c_j)$, tüm dokümanlar arasında cj sınıfının bulunma olasılığıdır.

Bayes bağımsızlık hipotezine göre $d_i = \{ w_1, w_2, \ldots w_n \}$ dokümanları için, dokümanları oluşturan tüm kelimeler "sözlük" adı verilen bir dosyada tutulur ve her sınıf için sözlük dosyası ile ortak olan kelimeleri için $P(d_i \setminus c_j)$ değeri hesaplanır:

$$P(d_i \setminus c_j) = \prod_{t=1}^{n} P(w_t \setminus c_j) \quad (4.5)$$

Buradan yola çıkarak NB sınıflandırıcısı (4.6) denklemindeki gibi basitleştirilebilir:

$$NB = \arg\max P(c_j) \prod_{t=1}^{n} P(w_t \setminus c_j)) \ , c_j \in C \quad (4.6)$$

(4.5) denkleminin hesaplanması aşağıdaki gibidir:

$P(w_t \setminus c_j)$: j. sınıfta w_t kelimesinin geçme olasılığı,

n_t : Kelimenin c_j sınıfında geçme sayısı,

n_j : Sınıf ile sözlükteki ortak kelime sayısı,

Sözlük : Kelimelerin birleştirilmesi ile elde edilen küme olmak üzere

$$P(w_t \setminus c_j) = \frac{1 + n_t}{n_j + |\text{Sözlük}|} \quad (4.7)$$

Örnek :

A = { The cat crabs the crolls off the stairs } = SINIF A = c_1

B= { It's raining cats and dogs } = SINIF B = c_2

C= {Cats eat mice and dogs bury bones } = TEST DOKÜMANI

1.Aşama :

Sınıflardaki kelimelerin birleştirilmesiyle elde edilecek olan sözlük kümesi oluşturulur:

Sözlük ={ cat, crab, croll, stair, rain, dog}

2.Aşama :

$P(c_1) = 1/2$, $P(c_2) = 1/2$

n = Sözlükteki kelime sayısı =6

n_1 = c_1 sınıfının kelime sayısı = 4

n_2 = c_2 sınıfının kelime sayısı = 3

3.Aşama **:**

Gelen test dokümanı ile sözlük dosyasının ortak kelimeleri bulunur. Örneğimize göre test dokümanının sözlük ile ortak kelimeleri = "cat" ve "dog" kelimeleridir.

4.Aşama :

Belirlenen ortak kelimeler ile olasılık hesaplarına dayanarak, testteki kelimelerin sınıflara aitlik yüzdesi hesaplanır.

$P(w_k/c_j) = \frac{n_{kj} + 1}{n_j + |\text{Sözlük}|}$ hesaplaması yapılır.

$n_{kj} = n_{kj}$ kelimesinin incelenen sınıfta kaç kez geçtiğini gösterir.

n_j = İncelenen sınıftaki toplam kelime sayısını gösterir.

|Sözlük| = Oluşturulan sözlük dosyasındaki kelime sayısını gösterir.

Testin A sınıfına ait olma olasılığı:

$$P(A) * P(\text{cat}/A) * P(\text{dog}/A) = \frac{1}{2} * \frac{1+1}{4+6} * \frac{0+1}{4+6} = 0.01$$

Testin B sınıfına ait olma olasılığı:

$$P(B) * P(\text{cat}/B) * P(\text{dog}/B) = \frac{1}{2} * \frac{2}{9} * \frac{2}{9} = 0.0247$$

Yukarıdaki sonuçlara göre olasılık yüzdesi yüksek çıkan sınıf B sınıfıdır. Bu durumda test dokümanı B sınıfına aittir.

4.3 Çoklu Naive Bayes Sınıflandırıcısı

Çoklu Nominal Bayes çalışmaları iki ciddi sorunun tespitiyle başlamıştır. Bu sorunlardan ilki kaba parametre tahminidir. İlk ÇNB modelinde bir sınıf için kelime olasılıkları, tüm pozitif eğitim dokümanlarının olabilirlikleri hesaplandıktan sonra tahmin edilmektedir. Yani, verilen bir sınıf için tüm deneme dokümanları birleştirilip bir büyük doküman içine konulmaktadır. Bu ise modeldeki parametre tahmininin, uzun dokümanlardan oluşan veri tabanında, kısalara göre daha fazla etkilenmesi anlamına gelmektedir. İkinci sorun ise sadece birkaç deneme dokümanı içeren seyrek kategorilerle uğraşırken ortaya çıkmaktadır. Naive Bayes gibi bir olasılık modelinin, yetersiz deneme örneğiyle iyi sonuçlar veremeyeceğini tahmin etmek zor olmaz. Eğer bir kategorinin sadece birkaç deneme dokümanı varsa, kategorinin ne olduğunu belirleyecek olan çok az sayıda bilgi verici kelime ve bunlara kıyasla çok fazla sayıda gürültülü terim bulunmaktadır. İşte ÇNB algoritması bu sorunları gidermek amacı ile oluşturulmuştur.

ÇNB, bir dokümanı kelimelerin çoklu dağılımına göre ele alır. Her doküman birbirinden bağımsız kelimelerin oluşturduğu metinler olarak değerlendirilir. Sınıflandırılmak üzere çoklu parametrelerin bir seti ile belirtilmiş sınıf kümesi c € {1,2,..m} olsun. Bu c sınıfı için tanımlanmış olan parametre vektörü $\vec{\theta}_c = \{ \theta_{c1}, \theta_{c2}, \ldots \theta_{cn} \}$ ile ifade edilsin. Burada n, sözlük kümesinin boyutudur. $\sum_i \theta_{ci} = 1$ ve θ_{ci}, i.kelimenin c sınıfında geçme olasılığıdır. Doküman için benzerlik olasılığı hesabı, kelimelerin dokümanlardaki olasılıklarını içeren θ parametrelerinin çarpımları ile bulunur.

$$P(d\backslash\vec{\theta}_c) = \frac{(\sum_i f_i)!}{\prod_i f_i!} \prod_i (\theta_{ci})^{f_i} \tag{4.8}$$

f_i, i.kelimenin d dokümanında geçme sayısını tutmaktadır. $P(\vec{\theta}_c)$ olasılığını sınıflar setine eklersek sınıflandırma için en yüksek olasılığın seçildiği sınıflayıcı aşağıdaki şekilde olur:

$$l(d) = \text{argmax}_c [\log P(\vec{\theta}_c) + \sum_i f_i \log \theta_{ci}] \tag{4.9}$$

$$\mathbf{l(d)} = \text{argmax}_c [b_c + \sum_i f_i w_{ci}], \tag{4.10}$$

b_c kelime sınır (treshold) değeri, w_{ci} ise c sınıfının i.kelime için olan ağırlık değeridir.

Sınıflama probleminde, sınıfların sayısı ve her sınıfın etiketlenmiş deneme verileri verilirken sınıfların parametreleri verilmemektedir. Parametreler, eğitim verilerinden tahmin edilmek zorundadır. Bir dokümandaki i. kelimenin c sınıfına ait dokümanlarda kaç kez geçtiğinin (N_{ci}), c sınıfındaki kelimelerin toplam sayısına (N_c) bölümü çoklu nominal parametrenin tahmini için basit bir biçim sunmaktadır.

$$\hat{\theta}_{ci} = \frac{N_{ci} + \alpha_i}{N_c + \alpha} \tag{4.11}$$

Burada α, α_i'lerin toplamıdır. α_i'ler her kelime için farklı olarak ayarlanabilecekken, burada bütün kelimelerde α_i=1 alınarak ortak bir uygulama sunulmaktadır.

Doğru parametreleri (4.15) denkleminde yerine koyarak, ÇNB sınıflayıcısını elde ederiz:

$$l_{ÇNB}(d) = \operatorname{argmax}_c [\log \hat{p}(\theta_c) + \sum_i f_i \log \frac{N_{ci} + \alpha_i}{N_c + \alpha}] \tag{4.12}$$

Burada $\hat{p}(\theta_c)$ kelime tahminlerine benzer biçimde tahmin edilebilir. Nitekim, sınıf olasılıkları, kelime olasılıklarının kombinasyonlarından aşırı derecede etkilenmeye meyillidir. Bundan dolayı kolaylık olması için düzgün bir $\hat{p}(\theta_c)$ tahmini kullanılmaktadır. Karar sınırları için olan ve ÇNB sınıflayıcısı tarafından tanımlanan ağırlıklar logaritma parametre tahminleridir:

$$\hat{w}_{ci} = \log \hat{\theta}_{ci} \tag{4.13}$$

4.4 Tamamlayıcı Naive Bayes Teoremi

NB'in birçok sistemik hatası mevcuttur. Sistemik hatalar, bir sınıfın diğerini uygunsuz biçimde himaye altına almasına neden olan yan etkilerdir. Bu bölümde, NB'in zayıf biçimde çalışmasına neden olan iki sistematik hata incelenecektir. Bunların yanlış sınıflamaya nasıl yol açtığı vurgulanıp bunları azaltmak ve ortadan kaldırmak için üretilen TNB çözümü üzerinde durulacaktır.

Çarpıtılmış veri (bir sınıf için bir başkasına göre daha fazla deneme örneği) sınıflandırma aleyhine etki edecek karar sınır ağırlıklarının varlığına yol açar. Bu durum sınıflayıcının, farkında olmadan, bir sınıfı diğerine kıyasla tercih etmesine sebep olur. Bu eğilimin nedenlerini incelerken ortaya atılan çalışmalar sonucu TNB kuralı doğmuştur.

Çizelge 4.3'de gösterilen ifade iki sınıflı basit bir sınıflama örneğidir. Her sınıf, binom dağılımına göre sırasıyla θ = 0.25 ve θ = 0.2 şeklinde tura (H) olasılığına sahiptir. Bize 1. sınıf için bir, 2. sınıf için iki deneme örneği verilmektedir ve bir turanın (H) ortaya çıkması etiketlenmek istenmektedir. Deneme verilerinin, bütün mümkün kümelerdeki maksimum olabilir parametreleri $(\theta_1 ve \theta_2)$ bulunacak ve bu parametreler turaların daha yüksek oranda ortaya çıkmasını tahmin eden sınıf ile test örneklerini etiketlemek için kullanılacaktır.

Tablo 4-3 İki sınıflı sınıflama örneği.

Class 1 $\theta = 0.25$	Class 2 $\theta = 0.2$	p(data)	$\hat{\theta}_1$	$\hat{\theta}_2$	Label for H
T	TT	0.48	0	0	*none*
T	{HT,TH}	0.24	0	$\frac{1}{2}$	Class 2
T	HH	0.03	0	1	Class 2
H	TT	0.16	1	0	Class 1
H	{HT,TH}	0.08	1	$\frac{1}{2}$	Class 1
H	HH	0.01	1	1	*none*

$$p(\hat{\theta}_1 > \hat{\theta}_2) = 0.24$$
$$p(\hat{\theta}_2 > \hat{\theta}_1) = 0.27$$

Tablo 4.3, yanlış eğilimin basit bir örneğidir. Bu örnekte, 1. sınıf 2.'ye göre daha yüksek oranda turaya (H) sahiptir. Ancak, bizim sınıflayıcımız 2. sınıf için 1.'ye göre daha sık tura ortaya çıkışı etiketlemiştir. 1. sınıfta daha düşük yazı oranı olmasına rağmen bir yazı örneğini 1. sınıf olarak daha sık etiketlemiştir. Bu durum, 2. sınıfın daha muhtemel varsayılmasından dolayı değildir. Sebep "çarpıtılmış veri eğilimleri" olarak tanımlanan etkidir. Bu etki dengesiz deneme verilerinin sonucu ortaya çıkmıştır. Eğer her sınıf için aynı sayıda deneme örneği kullanacak olursak beklenen sonuca, yani bir tura örneğinin daha yüksek tura oranına sahip bir sınıfla daha sık etiketlenmesi sonucuna ulaşırız.

Çarpıtılmış veriyle başa çıkmak için, Tamamlayıcı Naive Bayes adı verilen Naive Bayes'in bir "tamamlayıcı sınıf" sürümü ortaya konmuştur. Düzenli ÇNB ağırlıklarını (4.12) tahmin etmek için, tek bir c sınıfının deneme verileri kullanılmaktadır. Karşıt olarak TNB, parametreleri c dışındaki tüm sınıflardan veri kullanarak tahmin etmektedir. TNB tahminleri, ağırlık tahminindeki eğilimi azaltan sınıf başı deneme verisi kullandığından daha etkilidir. Böylece, ağırlık tahmini ve iyileştirilmiş sınıflama doğruluğunu daha kesin biçimde elde edebilmektedir.

TNB'nin tahmini:

$$\hat{\theta}_{\bar{c}i} = \frac{N_{\bar{c}i} + \alpha_i}{N_{\bar{c}} + \alpha} \tag{4.14}$$

Burada $N_{\bar{c}i}$, i. kelimenin c dışındaki sınıflardaki dokümanlarda kaç kez ortaya çıktığının sayısı iken, $N_{\bar{c}}$ c dışındaki sınıflardaki kelimelerin toplam ortaya çıkma sayısıdır. (4.11) denklemdeki gibi α_i ile α da düzleştirme parametreleridir. Önceden olduğu gibi ağırlık tahmini $\hat{w}_{ci} = \log \hat{\theta}_{ci}$ dir ve sınıflama kuralı da,

$$l_{TNB}(d) = \operatorname{argmax}_c [\log p(\vec{\theta}_c) - \sum_i f_i \log \frac{N_{\bar{c}_i} + \alpha_i}{N_{\bar{c}} + \alpha}] \tag{4.15}$$

Negatif işaret, tamamlayıcı parametre tahminleriyle zayıf biçimde eşleşen c sınıfı dokümanlara atama yapmak istediğimiz gerçeğini göstermektedir.

4.5 En Yakın K-Komşuluk Sınıflandırıcısı

Bu algoritma eldeki test dokümanının, mevcut olan tüm eğitim dokümanlarına olan benzerliğinin hesaplanmasına dayalıdır. Eğitim setindeki dokümanlar, n adet özelliğe sahip olan veriler olarak tanımlanmaktadır. Her eğitim dokümanı, n boyutlu uzayda bir noktayı temsil etmektedir. Kategorisi bilinmeyen bir test dokümanı sınıflandırılmak istendiğinde K-EYK sınıflandırıcısı, eğitim setindeki tüm dokümanlar arasında test dokümanına benzerlik uzaklığı en yakın olan K adet uzaklığı ve bu uzaklığın hangi sınıfa ait olduğunu belirlemektedir. En yakın komşuları bulduktan sonra, bu komşulardan kategorisi en çok olanın kategorisi test dokümanın sınıfı olarak alınmaktadır.

K-EYK sınıflandırıcısına uzaklık bazlı öğrenme algoritması da denilebilir. Dokümanlar arasındaki uzaklığı bulmak için farklı formüller geliştirilmiştir. Bunlardan biri Öklit Uzaklık Algoritması (Euclidean Distance)'dır. n boyutlu uzayda iki noktayı temsil eden dokümanlar $X_1 = (x_{11}, x_{12}, ..., x_{1n})$ ve $X_2 = (x_{21}, x_{22}, ..., x_{2n})$ şeklinde gösterilsinler. Bu durumda dokümanlar arasındaki uzaklık:

$$\text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{4.16}$$

Dokümanların sahip olduğu özellikler nümerik olmalıdır. Özellikler, özellik çıkarım

algoritmalarına göre hesaplanan değerlerdir. Her doküman n tane özelliğe sahiptir. Formüle göre, sırasıyla dokümanlardaki özellikler karşılıklı olarak çıkarılıp kareleri alındıktan sonra birikimli olarak toplanır. Toplamın da karekökü alınır.

K-EYK algoritmasında özellikler çok büyük değerler olabilir. Bu durumda değerleri normalize etmek, işlem karmaşıklığını azaltır. V özelliği normalize edilecek olursa aşağıdaki formül kullanılabilir:

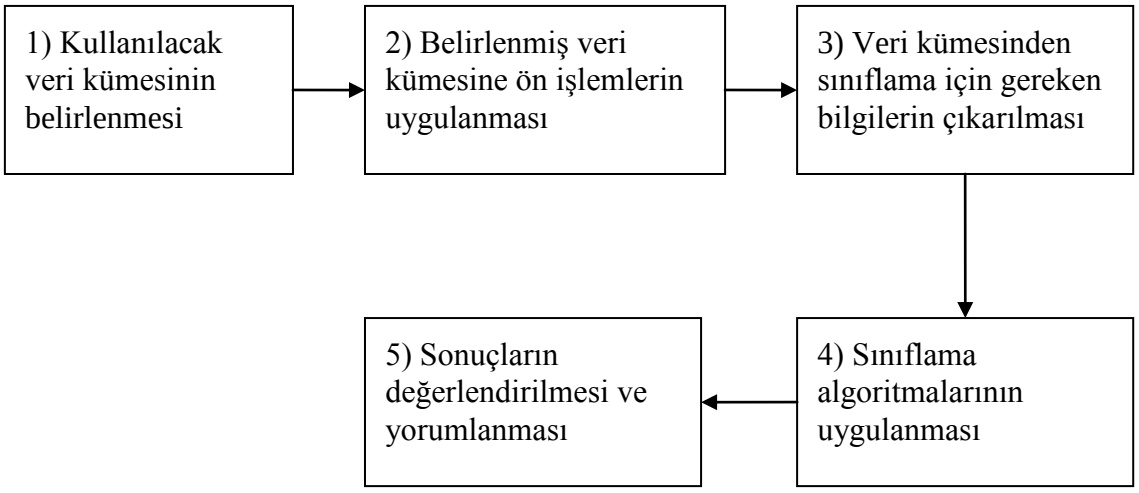
$$V' = \frac{V - \min_A}{\max_A - \min_A} \tag{4.17}$$

$\min_A$ ve $\max_A$ özellikler arasında en küçük ve en büyük olan değerlerdir.

K-EYK metoduna göre K değeri deneysel sonuçlara göre elde edilmektedir. Optimum K değeri, en küçük hata payının elde edildiği K değeridir. Bu değer, yazılan algoritmada K değeri değiştirilerek farklı denemeler yapıldığında elde edilir.

K-EYK sınıflandırılacak doküman sayısı fazla ise yavaş çalışır. | D| adet eğitim dokümanın olduğu bir veritabanında K=1 değeri için uzaklık hesabının karmaşıklığı 0(|D|) dir.

5. ALGORİTMALARIN UYGULANMASINDA İZLENEN AŞAMALAR



Şekil 5-1 Metin sınıflama algoritmalarının uygulanma aşamaları

Tez çalışmasında, metin sınıflama problemi Şekil 5.1'de belirtilen sıra ile incelenmiştir. İlk önce kullanılacak olan veri kümesi belirlenmiştir. Sınıflama algoritmalarının yüksek doğruluk oranlarına ulaşması, ön işlem aşamalarının ayrıntılı bir şekilde uygulanmasına bağlıdır. Bu nedenle, veri kümesini oluşturan dokümanlara metin sınıflama probleminde kullanılan tüm ön işlemler uygulanmıştır.

Ön işlem aşamasından sonra dokümanları temsil edecek olan bilgilere erişilmesi gerekmektedir. Bilgi erişimi, dokümanları oluşturan kelimelerin dokümanda geçme sıklığına bağlı olarak elde edilen ağırlık değerlerinin tespit edilmesi ile gerçekleştirilmektedir. Tespit edilen bilgiler, özellik uzayının boyutunu oluşturmaktadır. Bu boyutun büyük olması metin sınıflama probleminde karşılaşılan en önemli sorunlardandır. Boyut sorunun giderilmesi için dokümanlara özellik seçim teknikleri uygulanmıştır. Özellik seçim teknikleri ile uzay boyutu küçültülmüş ve veri kümesindeki dokümanlar sınıflama algoritmaları ile sınıflandırılmıştır.

5.1 Veri Kümesi Üzerinde Uygulanan Ön İşlemler

Şekil 5.1'de belirtilen sıra ele alınarak, önce üzerinde çalışılacak olan veri kümesi tespit

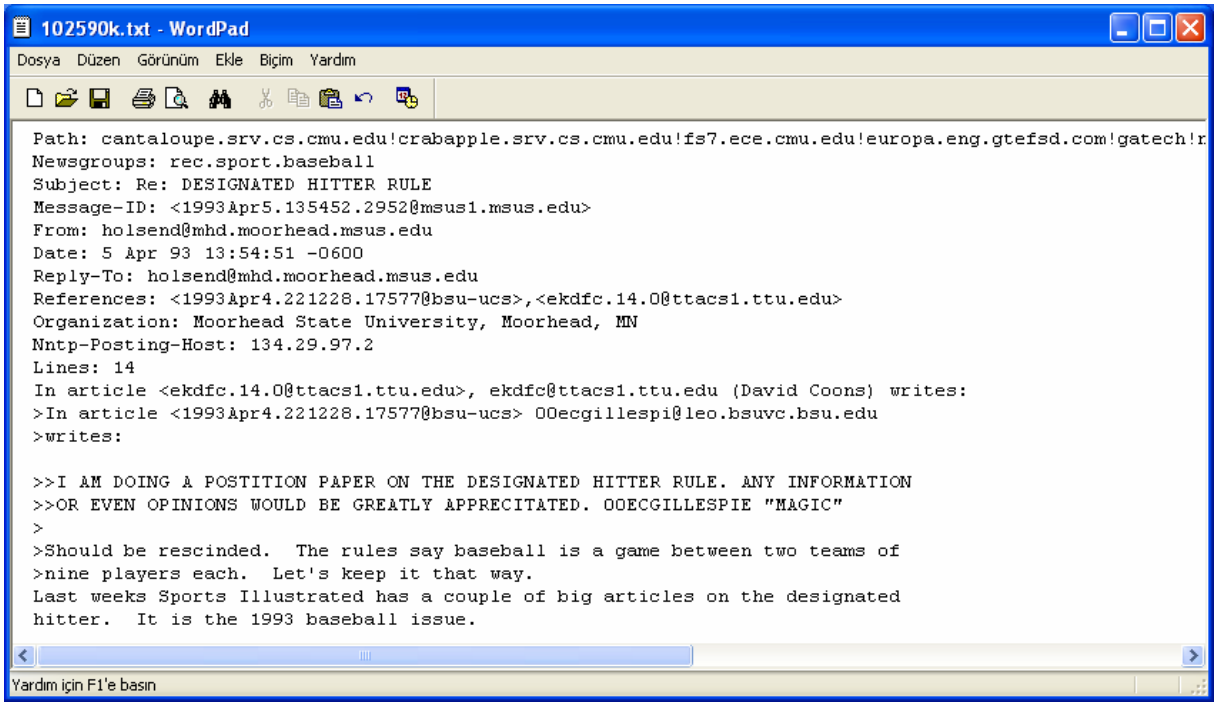
edilmiştir. Veri Kümesi, 20-Haber Grubu (20-Newsgroups)[**] veri kümesinin 1000 elemanlı bir alt kümesidir. Bu kümeyi oluşturan dokümanlar, birbirinden farklı konular hakkında yorum yapılan bir forum sitesinin metinsel verileri olarak toplanmışlardır. 20-Haber Grubu, 20 farklı kategoriye ait dokümanları içermektedir. Her bir kategoride 100'er adet doküman bulunmaktadır. Tez çalışmasında bu kategorilerden 10'u ele alınmıştır. İncelenen dokümanlar ve dokümanların ait oldukları kategoriler Tablo 5.1'de belirtilmiştir.

Tablo 5-1 Kullanılan veri kümesi ve kümenin ait olduğu kategoriler

Kategori	**Açıklama**	**Adet (Toplam =1000)**
alt.atheism	Ateizm ile ilgili konular	100
rec.autos	Otomobil ile ilgili konular	100
rec.sport.baseball	Beysbol ile ilgili konular	100
rec.sport.hockey	Hokey ile ilgili konular	100
rec.motorcycle	Motosiklet ile ilgili konular	100
sci.crypt	Kriptoloji ile ilgili konular	100
sci.electronics	Elektronik ile ilgili konular	100
sci.med	Sağlık ile ilgili konular	100
sci.space	Uzay ile ilgili konular	100
talk.politics	Politika ile ilgili konular	100

[**] Kullanılan veri kümesi http://kdd.ics.uci.edu/databases/20newsOKoups.data.html adresinden alınmıştır.

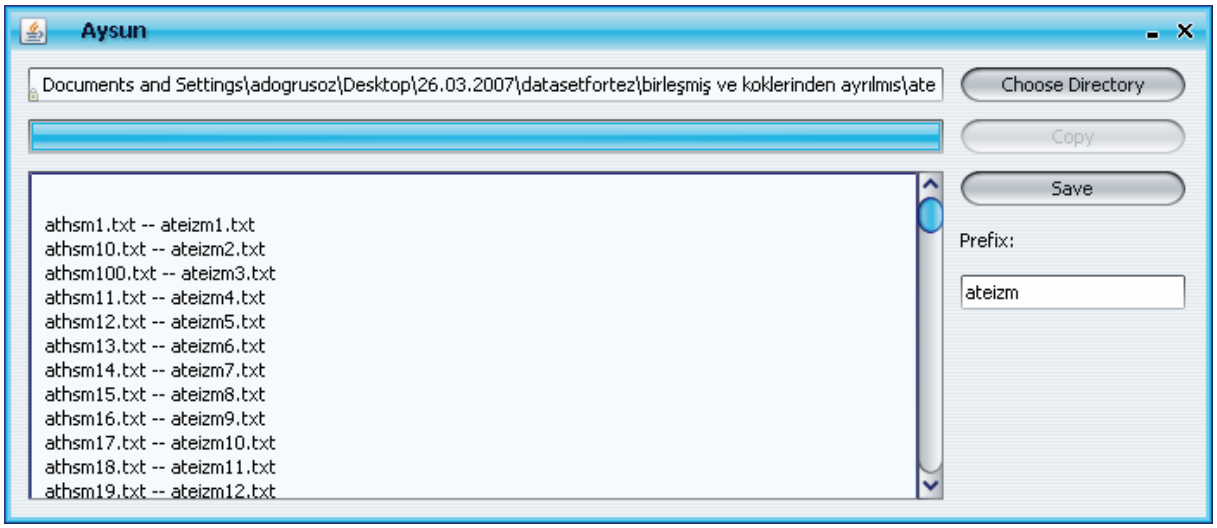
Çalışılacak veri kümesi belirlendikten sonra, veri kümesine bazı ön işlemler uygulanmıştır. Veri kümesine ait olan dokümanlar; dokümanın konusu, yazarı, tarihi ve referansı gibi bilgilerini içeren bir başlık kısmına sahiptir. Ön işlem aşamasında dokümanlar, bu başlık bilgilerinden arındırılmıştır. Başlık kısmını içeren örnek bir doküman Şekil 5.2'de görülebilmektedir.



Şekil 5-2 Başlık bilgisine sahip örnek bir doküman.

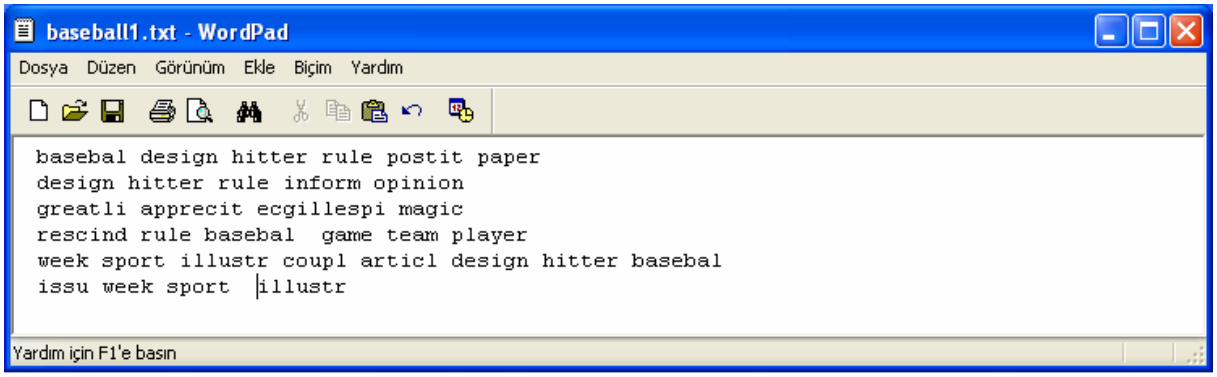
Ön işlem aşaması olarak yapılan bir başka işlem, dokümanlardaki noktalama işaretlerinin ve en önemlisi sınıflamada karakteristik özelliği olmayan kelimelerin dokümanlardan atılmasıdır. Gereksiz kelimeler ile kastedilen, İngilizcede sık geçen fiiller, zamirler, edatlar, bağlaçlar, rakamlar, gün, ay ve yıl isimleridir. Bu kelimeler, bir dosyada tutulmuş ve Java dili ile yazılan kaynak kod ile dokümanlardan arındırılmıştır. Kaynak kod ile dokümanın bulunduğu dosya yeri belirlendikten sonra dokümanlar sırası ile ön işlem aşamalarından geçirilmiş ve dokümanlara kategorilerinin isimleri verilmiştir.

Şekil 5.3'de görülen program çıktısı ile dokümanlar üzerinde ön işlem aşamaları uygulanmıştır. Aynı zamanda bu kod ile dokümandaki büyük harfle yazılmış olan kelimeler küçük harfe dönüştürülmüştür.



Şekil 5-3 Ön işlemleri gerçekleştirildiği Java dili ile kodlanmış çalışma.

Ön işlem aşamasında son olarak uygulanan aşama, dokümanları oluşturan kelimelerin köklerine ayrılmasıdır. Dokümanlardaki kelimeler, özellik uzayının boyutunun azaltılması amacı ile köklerine ayrılmıştır. Tez çalışmasında "Porter Stemmer" kök ayırma algoritması kullanılmıştır. Kök ayırma algoritması ile aynı köke sahip olan kelimeler bulunmuş ve özellik uzayının boyut büyüklüğü azaltılmıştır.



Şekil 5-4 Tüm ön işlem aşamalarının uygulandığı örnek doküman.

Sonuç olarak ön işlem aşaması, bilgi çıkarımı tekniklerinin uygulanması için gerekli ortamın hazırlandığı aşama olarak tanımlanabilir. Tez çalışmasında kullanılan veri kümesi üzerinde, metin sınıflama probleminde uygulanması gereken tüm ön işlem aşamaları uygulanmıştır. Tüm dokümanlar Şekil 5.4'de görülen biçime sokulmuştur ve bilgi çıkarımı için gereken duruma geçilebilecek seviyeye gelinmiştir.

5.2 Bilgi Erişim Teknikleri ve Sınıflama Algoritmalarının Uygulanması

Metin sınıflama probleminde dokümanların sınıflandırılması için dokümanları temsil edecek olan bilgilere erişmek gerekmektedir. Bilgi erişim sistemleri Bölüm 3'de ayrıntılı bir şekilde ele alınmıştır. "Cornell Smart" sistemine göre (Bkz Bölüm 3, Denklem 3.2) dokümanlardaki kelimelerin geçme sıklığı bilgisi kullanılarak, dokümanları karakterize eden iç temsilciler (kelime ağırlık değerleri) yaratılmaktadır. Bu değerlerden kelime sıklığı matrisi oluşturulmakta ve doküman temsilcilerine ulaşılmaktadır. Bilgi erişimindeki bu aşamalar, tez çalışmasında C dili kullanılarak yazılmış kaynak kod ile gerçekleştirilmiştir. Kaynak kodla ilgili bilgilere Ek-1'den ulaşılabilir.

Metin sınıflamada genel mantık, dokümanları oluşturan tüm kelimelerden bir kelime havuzu oluşturmaktır. Bu kelime havuzu, özellik uzayının boyutunu oluşturmaktadır. Veri tabanını oluşturan dokümanlarda geçen bütün kelimeler ayrı bir dosyada tutulmaktadır (sözlük dosyası). Amaç sözlük dosyasındaki kelimelerin her bir sınıftaki dağılımını bulmaktır. İşte gösterilen bu dağılımları bulmak için sınıflayıcı fonksiyonlar oluşturulmuş ve farklı sınıflama algoritmaları (NB, ÇNB, TNB ve K-EYK) ortaya konmuştur. Bu algoritmalardan bazıları istatistiksel yöntemlere (K-EYK), bazıları ise olasılık temelli yöntemlere (NB, ÇNB, TNB) dayanmaktadır.

Bilgi çıkarımından sonra tez çalışmasında sınıflama algoritmaları uygulanmıştır. Uygulamalar beş farklı durum altında gerçekleştirilmiştir. **İlk durumda** veri kümesine ait dokümanlardaki kelimeler köklerine ayrılmış ve dokümanların bilgileri çıkarılmıştır. Bu durumda özellik uzayı boyutunun oldukça fazla olduğu görülmüştür. Boyut büyüklüğünü azaltmak adına **ikinci durumda,** ilk durumda bulunan özellik uzayındaki kelimelerden toplamda sadece iki dokümanda geçenler kelime havuzdan silinmiş ve sınıflama algoritmaları bu koşul altında uygulanmıştır. **Üçüncü durumda** özellik seçim tekniklerinden "Bilgi Kazancı" tekniği ele alınmıştır. Özellik seçim algoritmaları ile özellik uzayının boyutu azaltılmıştır. Özellik seçim tekniklerinin kullanılması sınıflama oranlarını oldukça fazla bir oranda arttırmıştır. **Dördüncü durumda** özellik seçim tekniklerinden "Oransal Kazanç" tekniği ele alınmıştır. Son durum olan **beşinci durumda** ise "Ki-Kare" özellik seçim tekniği irdelenmiş ve sınıflama bu şartlar altında gerçekleştirilmiştir. Tüm bu durumlar "Weka Çalışma Platformu" kullanılarak uygulanmıştır. Bu platform ile ilgili bilgilere Ek-2'den erişilebilmektedir. Ele alınan beş farklı durum, uygulanan sınıflama algoritmalarına göre değişik başlıklar altında yorumlanacaktır.

5.2.1 Uzay Boyutunun Fazla Olduğu Durumun İrdelenmesi (1.durum)

İlk durumda dokümanlar sadece ön işlem aşamalarından geçirilmiştir. Veri tabanına ait olan dokümanlara özellik seçim teknikleri uygulanmamıştır. Dokümanlardaki kelimelerin oluşturduğu özellik uzayının kelime sayısı 12224'tür. Bu rakam 1000 dokümandan oluşan bir veri tabanı için oldukça büyük bir sayıdır. Boyut büyüklüğü dokümanlarda geçen tüm kelimelerin dikkate alınmasından kaynaklanmaktadır. Buradaki kelimelerin çoğu sınıfları karakterize edecek nitelikte değildir.

Sınıflama algoritmaları uygulanırken 10'lu Çapraz Geçerleme (10'lu ÇG) (10-Fold-Cross Validation) değerlendirme tekniği ele alınmıştır. Bu tekniğe göre her seferinde 1000 dokümanın yüzde onu test, geri kalanı eğitim dokümanları olarak kullanılmıştır. Her farklı test dokümanı için elde edilen doğruluk değerleri ve standart sapmalar Tablo 5.2 ve Tablo 5.3'de gösterilmiştir. Standart sapma değerlerinin incelenmesi daha doğru yorumların yapılmasını sağlayacaktır. Standart sapma değerinin düşük olması kullanılan test dokümanlarındaki doğruluk oranlarının değişkenliğinin az olması anlamına gelmektedir. Ayrıca tablolarda algoritmaların çalışma süreleri de görülmektedir. K-EYK'nın çalışma süresi NB'i temel alan sınıflayıcılara göre oldukça uzundur. Çünkü K-EYK'da, özellikler arası fark alınmaktadır.

NB teoremini temel alan sınıflayıcıların doğru sınıflama değerleri Tablo5.2'de görülmektedir:

Tablo 5-2 Birinci durumda NB sınıflayıcılarının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
NB	797	203	0,202	%79,7	06:45.4
ÇNB	912	88	0,122	%91,2	00:23.7
TNB	936	64	0,113	%93,6	00:25.7

Tablo 5.2'deki sonuçlara bakılacak olursa, doğru sınıflamada en iyi sonuca TNB ile ulaşıldığı görülmektedir. Bunun sebebi, TNB tekniğinin sınıflara ait kelime sayılarının homojen olmadığı durumlarda daha iyi sonuçlar vermesidir. ÇNB ve TNB algoritmalarının NB

algoritmasına göre daha iyi sonuç verme nedeni, tahmini parametre seçimlerinin birbirlerinden farklı olması ve bu tekniklerin dokümanları kelimelerin çoklu bileşenleri olarak ele almasıdır. Bu tekniklere ait olan formüller Bölüm 4'de ele alınmıştır.

K- EYK algoritmasının değişen K değerlerine göre değerleri Tablo 5.3'de belirtilmiştir:

Tablo 5-3 Birinci durumda K-EYK'nın doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi
K=1	345	655	0,360	%34,5	36:57.1
K=3	322	678	0,297	%32,2	37:02.3
K=5	299	701	0,296	%29,9	37:05.4
K=15	480	520	0,269	%48	37:09.5
K=35	262	738	0,288	%26,2	37:11.2

K değeri deneysel olarak belirlenmektedir. Standart sapması düşük, doğruluk yüzdesi yüksek olan K değeri, en uygun değer olarak belirlenmelidir. Tabloya göre bu durum K= 15 olma durumunda geçerlidir. K- EYK algoritmasının doğru sınıflama yüzdelerinin sonuçları, NB'i temel alan sınıflayıcılara göre oldukça düşüktür. K-EYK, özellik uzayındaki tüm kelimelerin ağırlık değerleri arasındaki farkı birikimli olarak toplar ve bir fark fonksiyonuna ekler. Bu nedenle K-EYK algoritması gereksiz kelimelerin varlığından olumsuz etkilenmektedir. Çalışma süresi, gereksiz kelimelerin varlığından ötürü uzundur ve algoritmanın doğruluk oranı aynı sebepten ötürü düşüktür.

5.2.2 Özellik Uzayının Boyutu Nasıl Azaltılabilir? (2.Durum)

Metin sınıflama probleminde en önemli sorun özellik uzayının boyut büyüklüğüdür. Sınıflamayı karakterize edecek olan kelimelerin önemsiz kelimelerden oluşması uzay büyüklüğünü arttırır. O halde sınıflama için ayırt edici özelliklere sahip olan kelimelerin seçilmesi boyut sorununu ortadan kaldıracak olan çözüm olabilir. Bu amaçla kelime havuzundaki kelimelerden, 1000 doküman arasından toplamda sadece iki dokümanda geçen

kelimeler silinmiştir. Bu koşul altında elde edilen özellik uzayı 4327 kelimeden oluşmaktadır. Değerlendirme yapılırken 10'lu ÇG tekniği kullanılmış ve alınan her farklı test seti için elde edilen doğruluk oranlarının standart sapma değerleri ile uygulanan NB, ÇNB, TNB ve K-EYK sınıflama algoritmalarının doğruluk oranları Tablo5.4 ve Tablo 5.5'de belirtilmiştir.

Tablo 5-4 İkinci durumda NB sınıflayıcılarının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
NB	795	205	0,202	%79,5	02:11.2
ÇNB	914	86	0,117	%91,4	00:07.7
TNB	931	69	0,113	%93,1	00:07.5

Özellik uzayının boyutunun azalması, iki tablodan da görüleceği gibi incelenen birinci duruma göre çalışma sürelerinin kısalmasını sağlamıştır. Aynı zamanda K-EYK algoritmasının sonuçlarının doğruluk yüzdelerini iyileştirmiştir.

Tablo 5-5 İkinci durumda K-EYK'nın doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
K=1	463	537	0,329	%46,3	13:37.5
K=3	328	672	0,296	%32,8	13:47.6
K=5	386	614	0,285	%38,6	13:48.4
K=15	386	614	0,269	%38,6	13:48.9
K=35	429	571	0,285	%42,9	13:49.3

K-EYK algoritmasında en yüksek doğruluk oranı ve en kısa çalışma süresi K=1 değeri için elde edilmiştir. Bu durumda K değeri bir olarak alınmalıdır.

5.2.3 Bilgi Kazancı Özellik Seçim Algoritmasının Uygulanması (3. durum)

Özellik seçim algoritmaları, sınıflama için önem derecesi yüksek olan kelimelere yüksek derecelendirme değerleri vermektedir. Kullanıcı, bu değerlerden bir alt sınır belirlemekte ve bu alt sınırın altında kalan kelimeleri özellik uzayından silmektedir. Belirlenecek olan alt sınır kullanıcıya bağlıdır. Bu tekniğin uygulanmasıyla özellik uzayının boyutu azaltılmış olmaktadır. Kelimelerin yüksek derecelendirme değerlerine sahip olması, bu kelimelerin doküman sınıflamada edindiği rolün fazla olması anlamına gelmektedir. Sınıflamada önemli rol üstlenen kelimelerin belirlenmesi sınıflama algoritmalarının da sonucunu olumlu biçimde etkileyecektir. Bilgi Kazancı tekniği ile özellik uzayı 394 kelimeden oluşmuştur. Özellik boyutunun azalması algoritmaların çalışma süresini önemli ölçüde azaltacaktır.

Bilgi Kazancı tekniğinde, kelimelere derecelendirme değerleri Tablo 4.1'de belirtilmiş olan ($\sum_{c\in\{c_i,\bar{c}_i\}}\sum_{t\in\{t_k,\bar{t}_k\}} P(t,c)\log_2\frac{P(t,c)}{P(t)P(c)}$) denklemi ile verilmektedir. Sınıflama algoritmaları uygulanırken 10'lu ÇG değerlendirme tekniği ele alınmıştır ve alınan her farklı test seti için elde edilen doğruluk oranlarının standart sapma değerleri belirlenmiştir. Özellik seçimi sonucu elde edilen sonuçlar Tablo 5.6'daki gibidir.

Tablo 5-6 Üçüncü durumda NB sınıflayıcılarının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
NB	942	58	0,107	%94,2	00:12.4
ÇNB	955	45	0,086	%95,5	00:02.3
TNB	950	50	0,100	%95	00:02.2

Tablo 5.6'dan görüldüğü gibi NB sınıflayıcılarının doğru sınıflama yüzdesi ilk iki duruma göre artmıştır, aynı zamanda çalışma süresi de azalmıştır.

Özellik seçim teknikleri K-EYK sınıflama algoritmasının sonuçlarını fazlaca yükseltmiştir. En yüksek doğruluk oranı K=1 olma durumunda geçerlidir. Sonuçlar Tablo 5.7 de görüldüğü gibidir:

Tablo 5-7 Üçüncü durumda K-EYK sınıflama algoritmasının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
K=1	939	61	0,105	%93,9	01:11.2
K=3	909	91	0,109	%90,9	01:12.6
K=5	927	73	0,106	%92,7	01:11.3
K=15	926	74	0,124	%92,6	01:11.8
K=35	908	92	0,174	%90,8	01:12.1

Kelime havuzundaki kelime sayısı en çok K-EYK sınıflayıcısını etkilemektedir. Bu duruma paralel olarak, özellik seçim tekniğinin kullanılması olumlu anlamda en çok K-EYK sınıflayıcısını etkilemiş ve çalışma süresinin de aynı olumlulukla kısalmasını sağlamıştır.

5.2.4 Oransal Kazanç Özellik Seçim Tekniğinin Kullanılması (4. durum)

Oransal Kazanç özellik seçim algoritması ile tıpkı Bilgi Kazancı tekniği gibi kelimelere derecelendirme değerleri verilmektedir. Derecelendirme büyük sayılardan oluşuyorsa bu algoritma derecelendirilen ifadeleri belli oranda küçültmektedir. Tekniğin uygulanması ile özellik uzayının boyutu 394 kelimeden oluşmuştur. Kelimelere derecelendirme değerleri verilirken Tablo 4.1'de belirtilmiş olan ($\frac{\sum_{c\in\{c_i,\overline{c}_i\}}\sum_{t\in\{t_k,\overline{t}_k\}} P(t,c)\log_2\frac{P(t,c)}{P(t)P(c)}}{\sum_{c\in\{c_i,\overline{c}_i\}} P(c)\log_2 P(c)}$) denklemi kullanılır. Elde edilen doğru sınıflama yüzdeleri Tablo 5.8 ve Tablo 5.9'da görülmektedir:

Tablo 5-8 Dördüncü durumda NB sınıflayıcılarının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
NB	943	57	0,105	%94,3	00:11.9
ÇNB	955	45	0,106	%95,5	00:02.1
TNB	943	57	0,087	%94,3	00:02.0

Veri kümesi üzerinde 10'lu ÇG değerlendirme tekniği uygulanmış ve her seferinde 1000 dokümanın %10'u test, geriye kalanı ise eğitim dokümanları olarak kullanılmıştır. Bu değişimde her farklı test için elde edilen doğruluk değerlerindeki sapmalara her iki tablodan ulaşılabilmektedir.

Bu tekniğin, BK tekniğinden farkı kelimelere verilen derece değerlerinin farklılık göstermesidir.

Tablo 5-9 Dördüncü durumda K-EYK sınıflama algoritmasının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
K=1	939	61	0,105	%93,9	01:08.7
K=3	909	91	0,109	%90,9	01:12.3
K=5	926	74	0,106	%92,6	01:12.4
K=15	926	74	0,124	%92,6	01:12.8
K=35	908	92	0,174	%90,8	01:13.1

Tablo 5.9'da görüldüğü gibi K değerinin bir olması durumda en yüksek doğruluk oranı elde edilmiştir.

5.2.5 Ki-Kare Özellik Seçim Tekniğinin Kullanılması (5. Durum)

Ki-Kare özellik seçim tekniği istatistiksel yöntemler kullanılarak oluşturulmuş olan bir tekniktir. Kelimelere derecelendirme değerleri Tablo 4.1'de belirtilmiş olan ($\frac{[P(t_k,c_i)P(\overline{t}_k,\overline{c}i)-P(t_k,\overline{c}_i)P(\overline{t}_k,c_i)]^2}{P(t_k)P(\overline{t}_k)P(c_i)P(\overline{c}_i)}$) denklemi ile verilmektedir. Değerlendirmeler yapılırken tüm durumlarda olduğu gibi 10'lu ÇG kullanılmış ve ele alınan farklı test setleri için bulunmuş doğruluk değerlerinin standart sapmaları tablolarda belirtilmiştir.

Özellik seçim algoritmalarının uygulanmasının doğurduğu olumlu sonuçlar bu teknikte de görülmektedir. Özellik seçim teknikleri ile boyutun azaltılması, birbirlerine yakın doğru sınıflama yüzdelerinin elde edilmesini sağlamıştır. Ki-Kare tekniğinde, Bilgi Kazancı ve Oransal Kazanç teknikleri ile neredeyse aynı sonuçlar elde edilmiştir. NB sınıflayıcılarının doğru sınıflama yüzdeleri ve 10'lu çapraz GÇ'deki standart sapma değerleri Tablo 5.10'da görülmektedir:

Tablo 5-10 Beşinci durumda NB teoremini temel alan sınıflayıcılarının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
NB	942	58	0,200	%94,2	00:12.0
ÇNB	955	45	0,129	%95,5	00:02.1
TNB	950	50	0,117	%95,0	00:02.0

Tablo 5.11'de görüldüğü gibi en iyi doğru sınıflama yüzdesi yine K=1 olma durumunda geçerlidir. Çalışma süreleri ve doğru sınıflama yüzdeleri NB sınıflayıcılarında olduğu gibi K-EYK sınıflayıcısında da diğer özellik seçim teknikleri ile yakın sonuçlar vermiştir.

Tablo 5-11 Beşinci durumda K-EYK sınıflama algoritmasının doğru sınıflama sonuçları.

	Doğru Sınıflanan Doküman Sayısı	Yanlış Sınıflanan Doküman Sayısı	10'lu ÇG'de S.Sapmalar	Doğru Sınıflama Yüzdesi	Çalışma Süresi(dk:sn)
K=1	939	61	0,326	%93,9	01:10.5
K=3	909	91	0,289	%90,9	01:12.8
K=5	927	73	0,278	%92,7	01:12.9
K=15	926	74	0,269	%92,6	01:13.1
K=35	908	92	0,281	%90,8	01:13.3

6. Algoritmaların Kıyaslanması

Metin sınıflama probleminde metinsel veri tabanlarının kullanılması doküman sayısının fazlalığına neden olur. Bu durum yüksek boyutlu bir özellik uzayının oluşumunu sağlar. Bundan dolayı dokümanlardaki kelimeler kullanılarak elde edilen bilgi çıkarımından sonra özellik uzayının boyutunu azaltmak adına özellik seçim teknikleri (BK, OK, Ki-Kare) uygulanmıştır. Kullanılan veri kümesinin özellik uzayının boyut büyüklüğü, özellik seçim tekniklerinin uygulanmasından önce ve sonra Tablo 6.1'de belirtildiği gibidir:

Tablo 6-1 Veri kümesine ait özellik uzayının farklı durumlar altındaki boyutu.

İncelenen Durumlar	**Özellik Uzayının Boyutu**
1. Durum (Kelimelerin köklerine ayrıldığı ve hiç bir özellik seçim tekniğinin uygulanmadığı durumdur.)	12224
2. Durum (Özellik uzayındaki kelimelerden toplamda 2 dokümandan az sayıda geçenlerin uzaydan silindiği durumdur)	4327
3. Durum (Özellik seçim tekniklerinden Bilgi Kazancı seçim tekniğinin uygulandığı durumdur)	394
4. Durum (Özellik seçim tekniklerinden Oransal Kazanç seçim tekniğinin uygulandığı durumdur)	394
5. Durum (Özellik seçim tekniklerinden Ki-Kare seçim tekniğinin uygulandığı durumdur)	394

Tablo 6.1'de görüldüğü gibi özellik seçim teknikleri, özellik uzayının boyutunu önemli ölçüde küçültmektedir. Buna paralel olarak, veri kümesine uygulanan sınıflama algoritmalarının efektifliğini yükseltmektedir. Farklı durumlar altında incelenen veri kümesine ait tüm doğru sınıflama yüzdeleri Tablo 6.2'de görüldüğü gibidir:

Tablo 6-2 Farklı durumlar altında incelenen veri kümesinin, farklı sınıflama algoritmalarına göre elde edilen doğruluk yüzdeleri.

	1.Durum	2.Durum	3.Durum	4.Durum	5.Durum
Tamamlayıcı Naive Bayes	%93,6	%91,3	%95,0	%95,0	%94,2
Çoklu Naive Bayes	%92,1	%91,4	%95,5	%95,5	%95,5
Naive Bayes	%79,7	%79,5	%94,2	%94,3	%95,0
K-EYK K=1	%34,5	%46,3	%93,9	%93,3	%93,9
K-EYK K=3	%32,2	%32,8	%90,9	%90,9	%90,9
K-EYK K=5	%29,9	%38,6	%92,7	%92,6	%92,7
K-EYK K=15	%48,0	%40,7	%92,6	%92,6	%92,6
K-EYK K=35	%26,2	%42,9	%90,8	%90,8	%90,8

Tablo 6.2'nin genelinden, K-EYK sınıflayıcısının doğru sınıflama yüzdesinin NB'i temel alan sınıflayıcılara göre daha düşük olduğu söylenebilir.

Tez çalışmasında ana amaçlardan biri, kelime havuzundaki boyut değişiminin sınıflama algoritmaları üzerindeki etkilerinin olumlu şekilde gösterilmesidir. Tablo 6.2, bu amacın gerçekleştirildiğinin ispatıdır. Özellik seçim tekniklerinin kullanılması (3, 4 ve 5. Durumlar) tüm sınıflama algoritmalarının performansını arttırmıştır.

Aşağıdaki tabloda (Tablo 6.3) ilk iki durumun sonuçları ele alınmıştır. Daha önce belirtildiği gibi ilk durum özellik uzayı boyutunun büyük olduğu, ikinci durum ise özellik uzayındaki kelimelerden ikiden az sayıda dokümanda geçen kelimelerin silindiği durumdur. Kelimelerin silinmesi özellik uzayının boyutunu azaltmıştır. Tablo 6.3'den görüleceği gibi ilk durumda uzay boyutu 12224 kelimeden oluşmuş, ikinci durumda ise bu boyut 4327'ye düşmüştür. Bu durumda önemli olan, boyut değişiminin sınıflama algoritmalarını nasıl etkilediğidir. İkinci durum incelendiğinde, K-EYK sınıflayıcısının sonuçlarının birinci duruma göre yükseldiği görülmektedir. Bu yükselmenin sebebi K-EYK sınıflayıcısının, sınıflamada karakteristik özelliği olmayan gereksiz kelime sayısından olumsuz anlamda çok fazla etkilenmesidir. Gereksiz kelime sayısının azaltılması bu nedenle K-EYK sınıflayıcısını iyileştirmiştir.

Tablo 6-3 Birinci ve ikinci durumun kıyaslanması.

	1.durum	2.durum
Özellik Uzayının Boyutu	**12224**	**4327**
TNB	%93,6	%91,3
ÇNB	%92,1	%91,4
NB	%79,7	%79,5
K=1	%34,5	%46,3
K=3	%32,2	%32,8
K=5	%29,9	%38,6
K=15	%48,0	%40,7
K=35	%26,2	%42,9

Yine Tablo 6.3'de görüldüğü gibi ikinci durumda NB'i temel alan sınıflayıcıların doğruluk oranlarının birinci duruma göre düştüğü görülmektedir. Bunun nedeni NB sınıflayıcılarının, kelimelerin her bir dokümanda geçme adedi temel alınarak hesaplanan olasılık değerlerini önemsemesidir. İki dokümandan az sayıda geçen kelimeler silindiğinde, kelimelerin dokümanda kaç adet geçtiği bilgisi önemsenmemiştir. Bu nedenle ikinci durumda, NB sınıflayıcılarının sonuçları olumsuz bir eğilim göstermiştir.

NB'i temel alan sınıflayıcıların performanslarını kıyaslamak için Tablo 6.4 incelenmelidir:

Tablo 6-4 NB teoremini temel alan sınıflayıcıların kıyaslanması.

	1. Durum	2. Durum	3. Durum	4. Durum	5. Durum
Özellik Uzayının Boyutu	**12224**	**4327**	**394**	**394**	**394**
TNB	%93,6	%91,3	%95,0	%95,0	%94,2
ÇNB	%92,1	%91,4	%95,5	%95,5	%95,5
NB	%79,7	%79,5	%94,2	%94,3	%95,0

NB teoremini temel alan sınıflayıcılar, Tablo 6.4'de görüldüğü gibi sınıflama probleminde her farklı durumda başarılı sonuçlar elde edilmesini sağlamıştır. ÇNB ve TNB sınıflayıcıları, NB

sınıflayıcısına göre daha iyi sonuçlar vermiştir. Bunun nedeni ÇNB ve TNB sınıflayıcılarının tahmini parametre seçimlerini farklı alması ve dokümanları kelimelerin çoklu bileşenleri olarak görmesidir.

TNB, çarpık veri eğilimini engeller. Özellik uzayının boyutu fazla iken (1. ve 2. Durum) Tablo 6.4'de gösterilen TNB sınıflayıcısı, ÇNB sınıflayıcısına göre daha iyi sonuçlara ulaşmıştır. Özellik seçim tekniklerinin uygulanması uzay boyutunu küçültmüştür. Bu nedenle çarpık veri eğilimi de azalmıştır. 3, 4 ve 5. Durumlar özellik seçim tekniklerinin uygulandığı durumlardır. Bu durumlar altında ÇNB sınıflayıcısı, TNB sınıflayıcısından daha yüksek sonuçlara ulaşmıştır.

Algoritmaların kıyaslanmasında incelenen bir başka durum çalışma süreleridir. Özellik uzayının boyut büyüklüğü çalışma süresini ters yönde etkiler. Tablo 6.5'da görüldüğü gibi özellik uzayını oluşturan kelimelerin sayısı azaldıkça algoritmaların çalışma hızı artar ve süre kısalır. Algoritmaların çalışma süreleri Windows XP Professional işletim sistemi ile çalışan, AMD Turion(tm) 64 X2 Mobile Technology TL-50 1.6GHz (2CPU) işlemcili, 894MB DDR-RAM'e sahip bir dizüstü bilgisayar ile hesaplanmıştır. NB teoremini temel alan sınıflayıcıların çalışma süresi K-EYK algoritmasına göre çok daha kısadır. Bunun sebebi K-EYK algoritmasının kelimelerin ağırlık değerleri arasındaki farkı temel almasıdır. NB, ÇNB ve TNB algoritmalarında NB algoritmasının çalışma süresinin daha fazla olduğu görülmektedir.

Tablo 6-5 Sınıflama algoritmalarının çalışma süreleri.

	1. Durum	2. Durum	3. Durum	4. Durum	5. Durum
ÖZELLİK UZAYININ BOYUTU	**12224**	**4327**	**394**	**394**	**394**
Tamamlayıcı Naive Bayes	00:25.7	00:07.7	00:02.2	00:02.0	00:02.0
Çoklu Naive Bayes	00:23.7	00:07.5	00:02.3	00:02.1	00:02.1
Naive Bayes	06:45.4	02:11.2	00:12.4	00:11.9	00:12.0
KNN k=1	36:57.1	13:37.5	01:11.2	01:08.7	01:10.5
KNN k=3	37:02.3	13:47.6	01:12.6	01:12.3	01:12.8
KNN k=5	37:05.4	13:48.4	01:11.3	01:12.4	01:12.9

7. SONUÇLAR VE ÖNERİLER

Tez çalışmasında, günümüzün popüler konularından olan metin sınıflama problemi her yönü ile ele alınmıştır. Dokümanlar öncelikle ön işlem aşamalarından geçirilmiştir. Bilgi çıkarımından sonra özellik seçim metotları ve sınıflama algoritmaları uygulanmıştır. Uygulanan algoritmaların sonucunda Bayes Teoremini temel alan sınıflayıcıların doğru sınıflama yüzdesinin daha fazla olduğu görülmüştür. K-EYK sınıflayıcısının doğru sınıflama yüzdesinin düşük olması sınıflama için kullanılan kelimelerin arasındaki uzaklık değerlerinin fazla olmasından kaynaklanmaktadır. Yani özellik uzayının boyutunun büyük olduğu durumlarda sınıflandırmayı olumsuz etkileyen kelime sayısı fazla olmaktadır. K-EYK algoritmasında özellik uzayına ait olan tüm kelimelerin ağırlıkları her bir dokümanın birbirlerine benzerlik oranının belirlenmesinde kullanılan fark fonksiyonuna eklenmektedir. Bu durum ise algoritmanın doğru sınıflama yüzdesinin düşük olmasına neden olur. Sınıflamanın iyileştirilmesi için özellik seçim metotları uygulanmıştır. Özellikle toplamda birden az sayıda geçen kelimelerin uzaydan silinmesi, sadece köklerin ayrılması durumuna göre daha yüksek doğrulama oranını ortaya koymuştur. Bu da K-EYK sınıflayıcısının özellik uzayı boyutundan fazlaca etkilendiğinin bir göstergesidir. Ayrıca K-EYK algoritmasında, K parametresi deneysel olarak belirlenmektedir. Yapılan denemelerde, K parametresinin en yüksek doğruluk oranını veren değerleri alınmıştır.

ÇNB ve TNB arasındaki fark ancak çarpık veri eğiliminin olduğu durumlarda bariz bir şekilde görülebilir. Çarpık veri eğilimi, dengesiz kelime dağılımlarına sahip dokümanlar üzerinde ortaya çıkan yanlış sınıflama sebebi olarak tanımlanabilir. TNB, çarpık veri eğilimini engellemek adına ÇNB'den farklı parametreler seçer. Doğru sınıflama oranları incelendiğinde özellik uzayının boyutunun fazla olduğu durumlarda TNB'in, ÇNB'den daha yüksek sonuçlar verdiği görülmüştür. Özellik seçimi ile boyutun küçültülmesi sonucunda ise ÇNB, TNB'den daha iyi sonuçlar sağlamıştır. NB sınıflayıcısı TNB ve ÇNB sınıflayıcılarına göre daha düşük sonuçlar vermiştir. Bunun sebebi ise TNB ve ÇNB sınıflayıcılarının, kelime sıklığı bilgisini dokümanları kelimelerin çoklu bileşenleri olarak ele alıp belirlemesidir. Bu nedenle TNB ve ÇNB, NB sınıflandırıcısına göre daha iyi sonuçlar sunmaktadırlar.

Özellik seçim metotlarının (BK, OK, Ki-Kare) uygulanması sadece K-EYK algoritmasının değil Bayes Teoremini temel alan diğer algoritmaların da daha iyi sonuçlar vermesini sağlamıştır. Özellik uzayının boyutunun azaltılması ile algoritmalar daha hızlı ve daha verimli sonuçlar ortaya koymuştur. Sonuçta özellik uzayının 394 kelimeden oluştuğu görülmüştür. Sadece köklerin ayrılması durumunda özellik uzayının boyutu 12224 kelimeden oluşmaktadır.

Özellik seçiminden sonra boyutun ne kadar çok azaldığı ve sınıflama performanslarının ne kadar çok arttığı ortadadır. O halde özellik seçim algoritmalarının sınıflama problemindeki temel sorunları giderdiği söylenebilir. Bu metotlardan en çok etkilenen sınıflayıcının da K-EYK sınıflayıcısı olduğu görülmüştür.

Özellik uzayının boyut büyüklüğü algoritmaların çalışma sürelerinin uzamasına sebep olmuştur. Uzay boyutu küçüldükçe çalışma süresi de kısalmıştır. K-EYK algoritması K parametresine bağlıdır ve ağırlık değerleri arasındaki farkın küçüklüğüne göre sınıflama yapmaktadır. Bu sebeplerden ötürü çalışma süresi NB sınıflayıcılarına göre oldukça fazladır.

Tez çalışmasında ortaya konan tüm bu çalışmaların incelenmesi sınıflama yüzdelerinin nasıl geliştirilebileceği konusunda bazı fikirlerin oluşumuna sebep olmuştur. Şu ana kadar yapılan çalışmalar incelendiğinde sınıflandırılacak dokümanlardaki kelime özelliklerinin birbirinden bağımsız olarak ele alındığı görülmektedir. Ancak bazı kelimeler, kelime gruplarıyla birlikte kullanılırsa sınıflamayı karakterize edecek olan bileşenler daha güçlü olabilir. Aynı zamanda kelimelerin anlamlarının ve türlerinin incelenmesi, eş anlamlı kelimelerdeki anlam farklılıklarının belirlenmesi ve sınıflamada bunların da ele alınması yine daha yüksek sonuçlar doğurabilir. Metinlerin gramer yapısının dikkate alınması da farklı bir bakış biçimi sunacaktır. Farklı yöntemlerin kullanılması, farklı özellik seçim metotlarının uygulanması ve bahsedilen diğer durumlar yeni çalışma alanları doğuracaktır. Tez çalışması bu alanda gerçekleştirilebilecek farklı çalışmalar için bir ön işlem aşaması olmuştur. Konu derinlemesine ele alınmış ve başarılı sonuçlar elde edilmiştir. Bundan sonraki çalışmalarda amaç önerilen yollarla sınıflama oranlarını yükseltebilmek olacaktır.

KAYNAKLAR

Domingos, P. ve Pazzani, M.J. (1997), "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss", Machine Learning, vol. 29, nos. 2/3, pp. 103-130.

Hall, M.A. ve Holmes, G. (2003), "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining", IEEE Transaction on Knowledge and Data Engineering, Vol.15, No.6. pp. 1-16.

Jiang, J., Shen, Y. (2003), "Improving the Performance of Naive Bayes for Text Classification", CS224N Spring, pp. 1-10.

Kim, S.B., Han, K.S., Rim, H.C. ve Myaeng S.H. (2006), "Some Effective Techniques for Naive Bayes Text Classification", IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 11., pp. 1457-1465.

Lewis, D.D. (1992), "Represantation and Learning in Information Retrieval", PhD dissertation, Dept. of Computer Science, Univ.of Massachusetts, Amherst.

Lewis, D.D. (1998), "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval", Proc.ECML-98, 10th European Conf. Machine Learning, C. Nedellec and C. Rouveirol, eds., pp. 4-15.

McCallum, A.K. ve NBKam, K. "Employing EM in Pool-Based Active Learning for Text Classification", Proc.ICML-98, 15th Int. Conf. Machine Learning, J.W.Shavlik, ed., pp.350-358.

NBKam, K., McCallum, A.K., Thrun, S. ve Mitchell, T.M. (2000), "Text Classification from Labeled and Unlabeled Documents Using EM", Machine Learning, vol.39, nos. 2/3, pp. 103-134.

Rennie, J.D.M., Shih, L., Teevan, J. ve Karger, D.R. (2003), "Tackling the Poor Assumption of Naive Bayes Text Classifiers", Proceeding of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC., pp.1-8.

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y ve Wang, Z. (2007), "A Novel Feature Selection Algorithm for Text Categorization", Expert Systems With Applications 33, pp. 1-5.

Soucy, P., ve Mineau, G.W. (2001), "A Simple KNN Algorithm for Text Categorization", IEEE 0-7695-1119-8, pp.647-648.

EKLER

Ek-1 Bilgi Çıkarımı İçin Yazılan C Kodu

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include <stdlib.h>
#include <malloc.h>
#include <conio.h>

#define     INWORD                          1
#define     OUTWORD                         0
#define     CLASS                           10
#define     TRAINSAYISI                     1000

FILE *voc,*weightss;

char sinif[CLASS][20] ={"1","2","3","4","5","6","7","8","9","10"};
char sinif2[TRAINSAYISI][20] =
{"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1
","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","
1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1"
,"1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1","1
","1","1","1","1","1","1","2","2","2","2","2","2","2","2","2","2","2","2","2","2","
2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2",
"2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2"
,"2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2
","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","2","
2","2","2","2","2","2","2","2","2","2","2","2","2","2","3","3","3","3","3","3",
"3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3"
,"3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3
","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","
3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3",
"3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3","3"
,"4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4
","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","
4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4",
"4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4"
,"4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4","4
","4","4","4","4","4","4","4","5","5","5","5","5","5","5","5","5","5","5","5","5","
5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5",
"5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5"
,"5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5
","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","5","
5","5","5","5","5","5","5","5","5","5","5","5","5","5","6","6","6","6","6","6",
"6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6"
,"6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6
","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","
6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6",
"6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6","6"
,"7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7
","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","
7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7",
"7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7"
,"7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7","7
","7","7","7","7","7","7","7","8","8","8","8","8","8","8","8","8","8","8","8","8","
8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8",
"8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8"
,"8","8","8","8","8","8","8","8","8","8",
```

```
"8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8"
,"8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8","8
","8","8","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","
9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9",
"9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9"
,"9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9
","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","9","
9","9","9","9","9","9","9","9","9","10","10","10","10","10","10","10","10",
"10","10","10","10","10","10","10","10","10","10","10","10","10","10","10",
"10","10","10","10","10","10","10","10","10","10","10","10","10","10","10",
"10","10","10","10","10","10","10","10","10","10","10","10","10","10","10",
"10","10","10","10","10","10","10","10","10","10","10","10","10","10","10",
"10","10","10","10","10","10","10","10","10","10","10","10","10","10","10",
"10","10"};

//trainWrdCounter: herbir train dokumanının kac adet kelime icerdigini
tutuyor.
//testWrdCounter: herbir test dokumanının kac adet kelime icerdigini
tutuyor.
//vocWrdCounter     : voc dosyasının kac adet kelime icerdigini tutuyor.
int trainWrdCounter[TRAINSAYISI],vocWrdCounter;

//Dosyalara ait kelime ozelliklerinin saklanacağı struct yapısı dizi olarak
tanımlandı.
struct ORTAK
{
      char name[30];
      int train_adet[TRAINSAYISI];

      int index;
      double train_weight[TRAINSAYISI];

}*word;
//voc dosyasının yaratılması fonksiyon olarak tanımlandı.
void openVocabulary(){

      if((voc=fopen("Vocabulary.txt","w+"))==NULL){
            fprintf(stderr, "Cannot open vocabulary.txt.\n");
      exit(1);
      }

}
//Vocabulary dosyası olusturuluyor...
//Train dosyalarına ait her kelime bir kez bu dosyaya ekleniyor...
void vocaKelimeEkle(FILE *file_name){
      char str[30],str_voc[30];
      int flag;
      while(fscanf(file_name,"%s",str)!=EOF){
                  flag=0;
                  while(fscanf(voc,"%s",str_voc)!=EOF){
                        if(strcmp(str,str_voc)==0){
                              flag=1;
                              break;
                        }
                  }
                  if(flag==0)
                        fprintf(voc,"%s\n",str);
                  rewind(voc);
            }
      rewind(file_name);
      rewind(voc);
```

```
}

//Kelime sayilarinin hesaplanmasi icin gereken fonksiyon parcası.
int issep(int ch)
{
	return isspace(ch) || ch == '.' || ch == ',' || ch == ';' || ch == '?' ||
		ch == '!' || ch == ':';
}
//Bir dokuman için kelime sayısı hesaplanıyor...
int Count_AllWords(FILE *file_name){
	int ch;
	int flag = OUTWORD;
	int wcounter = 0;
	rewind(file_name);
	while ((ch = fgetc(file_name)) != EOF){
		if (issep(ch))
			flag = OUTWORD;
		else if (flag == OUTWORD) {
			++wcounter;
			flag = INWORD;
		}
	}
	rewind(file_name);
	printf("%4d",wcounter);
	return wcounter;
}
// Algoritmada amac vocabulary dosyası ile ortak olan kelimelerin agırlıklarını
bulmaktır.
// Bu amacla vocabularydeki kelimeler word[k].name de saklandı.
// Bu kelimelere index degeri verildi ve bu word[k].index de tutuldu.
// kelimelerin dokumanlarda kackez gectigini tutmak uzere word[k].trainadet = 0  &
word[k].testadet = 0

void Find_Words(){
	char str_voc[30];
	int k=0,i;

	rewind(voc);
	while(fscanf(voc,"%s",str_voc)!=EOF){
		strcpy(word[k].name,str_voc);
		word[k].index=k+1;
		for(i=0;i<TRAINSAYISI;i++){
			word[k].train_adet[i]=0;  //i.train dokumandaki k. kelime
		}

		k++;
	}
	rewind(voc);
}
int main()
{
	int i,j,train_DF,test_DF,z;
	char str[30],fstr[30];
	double tf,idf;
	FILE *ftrain, *rtrain, *fwords;
	//train dosya isimleri dosyaisimleri.txtden okunuyor...
	if((rtrain=fopen("dosyaisimleri.txt","r"))==NULL){
		fprintf(stderr, "Cannot open dosyaisimleri.txt.\n");
	exit(1);
	}

	if((fwords=fopen("kackezgecti.txt","w"))==NULL){
		fprintf(stderr, "Cannot create 2den_az_olan_kelimer.txt.\n");
```

```
exit(1);
}

openVocabulary();

printf("\nVocabulary dosyasi yaratildi");

printf("\nVocabulary  dosyasi olusturuluyor ve Train dokumanlarinin
sozcuk sayilari hesaplaniyor...");


// Vocabulary dosyası olusturuluyor ve train dokumanların sözcük sayıları
hesaplanyor...
// Vocabulary dosyası train dokumanlarının birleştirilmesiyle olusmustur.
z=0;
while(fscanf(rtrain,"%s",fstr)!=EOF){

      if((ftrain=fopen(fstr,"r"))==NULL){
            fprintf(stderr, "Can not open files %d\n",z);
            exit(1);
      }
      vocaKelimeEkle(ftrain);
      printf("\ntrain[%d]:  ",z);
      trainWrdCounter[z]=Count_AllWords(ftrain);
      z++;
      printf("\t");
      Count_AllWords(voc);

      fclose(ftrain);
}

printf("\nvocabulary kelime sayisi:  ");
vocWrdCounter=Count_AllWords(voc);

//struct ORTAK için gerekli alaln tahsis ediliyor...

if((word=(structORTAK*)malloc(vocWrdCounter*(((TRAINSAYISI))*sizeof(int)+200*
sizeof(int)+(sizeof(double)*((TRAINSAYISI))))))==NULL){
      printf("\nYetersiz Bellek \n");
      exit(1);
}


//struct için gerekli atamalar yapılıyor...
Find_Words();
printf("\n ozelliklerin boyutu belirlendi");
z=0;
rewind (rtrain);
while(fscanf(rtrain,"%s",fstr)!=EOF){

      if((ftrain=fopen(fstr,"r"))==NULL){
            fprintf(stderr, "Cannot open files%d\n",z);
            exit(1);
      }

      while(fscanf(ftrain,"%s",str)!=EOF){
            for(j=0;j<vocWrdCounter;++j)
                  if(strcmp(word[j].name,str)==0)
                        word[j].train_adet[z]=word[j].train_adet[z]+1;
      }
      z++;
      fclose(ftrain);
}
```

```
printf("\nKelimelerin her train dokumanında geçme adedi bulundu");

for(j=0;j<vocWrdCounter;++j)
      for(i=0;i<TRAINSAYISI;i++)
            fprintf(fwords,"%d,",word[j].train_adet[i]);


//Agırlıklar hesaplanıyor...
for(j=0;j<vocWrdCounter;++j){
      train_DF=0;
      test_DF=0;
      for(i=0;i<TRAINSAYISI;i++){
            if(word[j].train_adet[i]!=0)
                  train_DF++;
      }

      printf("%d.kelime %d dokumanda gecti\n",j,train_DF);

      for(i=0;i<TRAINSAYISI;i++){
            if(word[j].train_adet[i]==0){
                  word[j].train_weight[i]=0;
            }

            else{
            tf = 1+log10(1+log10( word[j].train_adet[i]));

            idf=log10((double)(1+TRAINSAYISI)/(double)train_DF);
            word[j].train_weight[i]=tf*idf;

            }
      }

}
printf("\nAgirliklar belirlendi" );


if((weightss=fopen("weightss.txt","w"))==NULL){
      fprintf(stderr, "Cannot create weightss.txt.\n");
exit(1);
}
fprintf(weightss,"@RELATION weightss\n\n");
for(j=0;j< vocWrdCounter;++j){
      fprintf(weightss,"@ATTRIBUTE %s NUMERIC\n",word[j].name );
}
fprintf(weightss,"@ATTRIBUTE class {");
for(i=0;i<CLASS;i++){
      fprintf(weightss,"%s,",sinif[i]);
}
fprintf(weightss,"}\n@DATA\n");
for(j=0;j<TRAINSAYISI;++j){
      for(i=0;i<vocWrdCounter ;i++){
            fprintf(weightss,"%lf,",word[i].train_weight[j]);
      }
      fprintf(weightss,"%s\n",sinif2[j]);
}

return 0;

}
```

Ek-2 Weka Çalışma Platformu

Weka çalışma platformu, Waikato üniversitesinde yapılan çalışmalar sonucu kaynak kodu Java dili ile yazılmış makine öğrenmesi algoritmalarını içermektedir. Sınıflama ve bilgi çıkarım algoritmaları "Explorer" kullanıcı ara yüzünde bulunmaktadır.

Şekil Ek 2.1 Weka çalışma platformu.

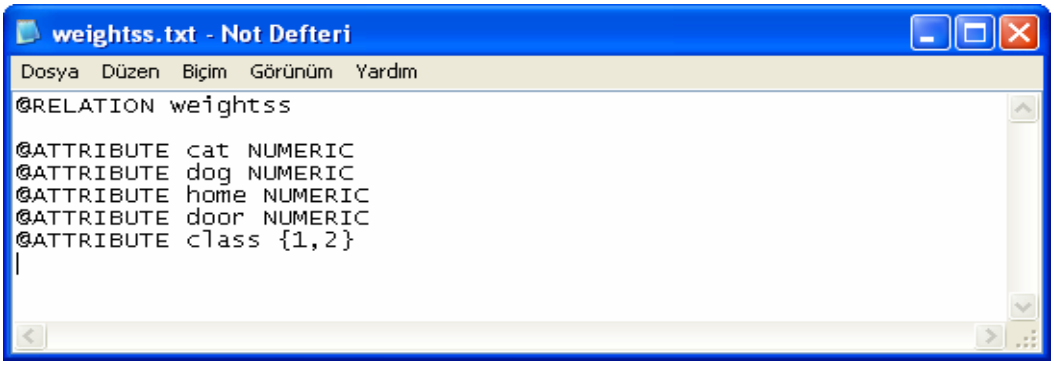
Weka kullanıcı ara yüzünde dokümanları temsil edecek olan bilgiler ARFF uzantılı dosyalarda saklanmaktadır. ARFF (Attribute – Relation File Format) uzantılı dosyalar verileri özellikleriyle birlikte barındıran ASCII doküman dosyalarıdır. Bu formatta dosyaların iki ayrı kısmı mevcuttur. Birinci kısım başlık bilgilerinin, ikinci kısım ise veri bilgilerinin bulunduğu kısımdır. Başlık bilgileri dosya ismini ve özellik tanımlamalarını içermektedir. Dosya ismi ile ilişki kurmak için ARFF dosyasının ilk satırına yazılması gereken bilgi formatı **@relation < dosya adı>** şeklindedir. Özellik tanımlamaları için kullanılan format ise **@attribute <data ismi> <data tipi**> şeklindedir. Data ismi alfabetik bir karakterle başlamalıdır. Eğer isim boşluk karakteri içerecekse bu durumda tırnak içerisine alınmalıdır.

ARFF dosyasında geçerli olan veri tipleri:

- Reel sayıları içeren veri tipi (numeric data type)
- Elle girilen özellik isimlerini içeren veri tipi (nominal specification)
- Yazısal değerleri içeren veri tipi (string data type)

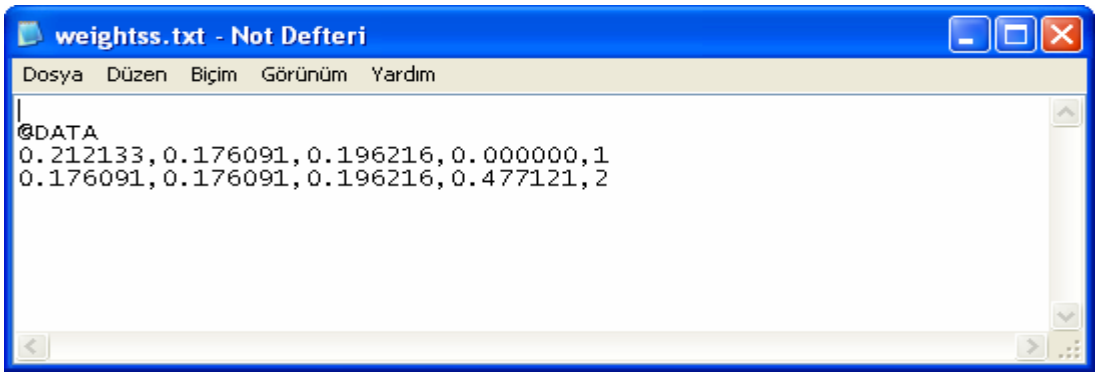
Dokümanları temsil edecek olan bilgilerin ve bilgi türlerinin bulunduğu ARFF uzantılı

dosyanın başlık kısmı aşağıdaki şekilde görülmektedir. Her bilgi verilerin ağırlık değerlerinin bulunacağı kolonları ayırmak için kullanılmaktadır.



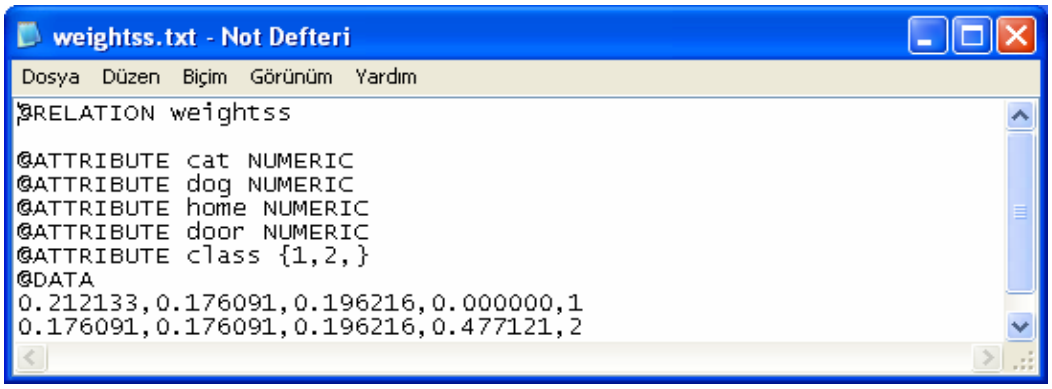
Şekil Ek 2.2 Arff uzantılı dosyanın başlık kısmı.

Bilgiler Cornell Smart Sistemine göre hesaplanan ağırlık değerleridir. Bu ağırlık değerlerinin çalışma platformunda kullanılır hale gelebilmesi için başlık bilgilerinin hemen altına **@DATA** bilgisi yazılmalıdır. Bu tanımlamadan sonra ağırlık değerleri dosyaya yazdırılmalıdır.



Şekil Ek 2.3 Arff uzantılı dosyanın veri bilgileri kısmı.

Sonuçta başlık bilgileri ve ağırlık değerlerinin bulunduğu ARFF uzantılı dosya örneği aşağıdaki şekilde görüldüğü gibidir:



Şekil Ek 2.4 Arff uzantılı dosya örneği.

ÖZGEÇMİŞ

Doğum tarihi	17.06.1983	
Doğum yeri	Kadiköy	
Lise	1998 – 2001	Kadir Has Anadolu Lisesi
Lisans	2001 – 2005	Yıldız Üniversitesi Kimya - Metalürji Fak. Matematik Mühendisliği Bölümü
Yüksek Lisans	2005 – 2007	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı

Çalıştığı kurumlar

2005 – devam	Doğuş Üniversitesi Bilgisayar Mühendisliği Araştırma Görevlisi