KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

WEB İÇERİK MADENCİLİĞİ VE KONU SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

FATİH GÜRCAN

TEMMUZ 2009
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

WEB İÇERİK MADENCİLİĞİ VE KONU SINIFLANDIRILMASI

Fatih GÜRCAN

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"Yüksek Lisans (Bilgisayar Mühendisliği)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 10.06.2009
Tezin Savunma Tarihi : 03.07.2009

Tez Danışmanı : Yrd. Doç. Dr. Cemal KÖSE

Jüri Üyesi : Prof. Dr. Vasif V. NABİYEV

Jüri Üyesi : Prof. Dr. Sefa AKPINAR

Enstitü Müdürü : Prof. Dr. Salih TERZİOĞLU

Trabzon 2009

ÖNSÖZ

İnternet, bilgi paylaşımı ve hayatımıza getirdiği kolaylıklar açısından, günümüzde bir çok bilim adamı tarafından yüzyılın en büyük icadı olarak gösterilmektedir. İnternet teki kullanışlı bilgiye erişim ve bilginin doğru bir şekilde kategorize edilmesi, bilgisayar bilimlerinin en önemli problemlerinden biri olarak karşımıza çıkmaktadır. Böylesine güncel ve faydalı bir tez konusu seçiminde bana yol gösteren, benden hiç bir zaman yardımını esirgemeyen Sayın Danışman Hocam Yrd. Doç. Dr. Cemal KÖSE' ye, yardım ve desteğinden dolayı sonsuz teşekkür ve şükranlarımı sunarım.

Tez çalışmalarımda desteklerini esirgemeyen Bilgisayar Mühendisliği bölümündeki hocalarıma teşekkür ederim.

Tez çalışmalarımda yardım ve katkılarından dolayı Müslüm ÖZTÜRK, Sinan AKYAZICI, Yasin KAYA, Ercüment YILMAZ, Özkan BİNGÖL ve Uğur ŞEVİK' e teşekkürlerimi sunarım.

Öncelikle, beni yetiştirip bu günlere getiren sevgili Annem Çiçek GÜRCAN ve Babam H.Hüseyin GÜRCAN' a, diğer aile fertlerime ve arkadaşlarıma, saygı ve sevgilerimi sunarım.

Ayrıca, çalışmaya emeği geçen, ismini yazamadığım tüm arkadaşlarıma ve Karadeniz Teknik Üniversitesi' ne teşekkürlerimi sunarım.

Fatih GÜRCAN
Trabzon 2009

İÇİNDEKİLER

Sayfa No

ÖZET

İnternet çok büyük bir bilgi deposudur. İnternetteki bu bilgiler büyük olduğu kadar düzensiz ve birbirinden bağımsız oluşturulmuş bilgilerdir. Bu yönüyle web deki bilgiler tamamen, anlamlı ve işe yarayan bilgiler değildir. Bu büyük düzensiz verilerden anlamlı bilgilerin elde edilebilmesi için, günümüze kadar değişik metotlar denenmiştir. Web İçerik Madenciliği, World Wide Web deki bütün dokümanları (metin, resim, ses, görüntü v.s.) inceleyerek, bu dokümanların içerikleri arasındaki ilişkisel benzerlikleri ve farklılıkları ortaya çıkaran bir metottur. Böylece birbiriyle gerçek anlamda ilişkili ve aynı konuda olan sayfalar, kendi içinde sınıflandırılabilir. Sayfaların içeriği analiz edilir ve sayfanın temeline inilerek gerçekte sayfanın hangi konuyu içerdiğine bakılır. Bu çalışmada, web ortamları için, Google arama motoru ile bütünleşik, bir konu sınıflandırma sistemi geliştirilmiştir.

Ayrıca metin sınıflandırma da kullanılan Navie Bayes, Destek vektör makinası, K-en yakın komşuluk algoritması ve karar ağacı algoritmalarının sınıflandırma performansı test edilmiş ve sonuçlar karşılaştırılmıştır.

Yapılan analiz sonucunda sayfanın gerçekte hangi konu ile ilgili olduğu tahmin edilmiştir. Yapılan bu tahminlerin, web ortamında, kullanıcıların aradığı bilgilere daha kestirme ulaşmasına yardımcı olacağı düşünülmektedir.

Anahtar Kelimeler: Web İçerik Madenciliği, Arama Sonuç Madenciliği, Metinsel Veri Madenciliği, Bilgi Çıkartımı, Konu Sınıflandırma, Metin Sınıflandırma Algoritmaları.

SUMMARY

Web Content Mining and Subject Classification

Internet is an enormous information resource. The vast amount information on the internet is unsystematic and independent from each other as well. This information is not also meaningful and usable in this respect. Several methods have been applied to obtain meaningful information from this disordered data accumulation. Web content mining is a method that discovers similarities and differences between those documents such as text, picture, video etc by analyzing them. In this manner, documents and pages which are truly related and about the same subject can be classified.

Hence, the contents of pages are analyzed and the real content of the pages are categorized. In this study, the pages are classified by taking into account certain criteria, and results of the classification which subjects of the pages, are determined related to the real content of the pages. Text categorization techniques which are used in this study (Naive Bayes, K- Nearest Neighbor, Support Vector Machine and Decision Trees) are examined on web documents for classification of the subject of the documents. These techniques have also been compared with each other. Thus, the web users may utilize these results to get directly aimed information in search of data.

Key Words: Web Content Mining, Search Result Mining, Textual Data Mining, Information Extraction, Subject Classification, Text Classification Algorithms.

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

1. GENEL BİLGİLER

Günümüzde her alanda küresel olarak, erişilebilir veri miktarı büyük bir hızla artmaktadır. Özellikle, web ortamında paylaşılan veri miktarında, son yıllarda büyük bir patlama yaşanarak tahminlerin çok üzerinde bir artış olmuştur. Hatta bu oran, her yıl ikiye katlanarak artış büyük bir hızla devam etmektedir. Günümüzde, yaklaşık olarak 300 milyon internet sayfası olduğu tahmin edilmekte ve her gün 1 milyon sayfa daha eklenmektedir [20]. Aynı şekilde veri tabanlarının sayısı da benzer, hatta daha yüksek bir oranda artmaktadır. Buna birde günümüzde yaygınlaşan elektronik ticaret, online alışveriş, portal ve iletişim sistemlerinin de eklenmesiyle veri madenciliği kavramı daha da ön plana çıkmıştır.

Bilgisayar sistemleri ile üretilen bu veriler tek başlarına değersizdirler (Özellikle veri tabanlarının bilgiyi sadece saklamak için dizayn edildiği düşünüldüğünde). Çünkü çıplak gözle bakıldığında verilerin bir anlam ifade etmediğini söyleyebiliriz. Bu veriler belli bir amaç doğrultusunda işlendiği zaman anlamlı hale gelmektedir [22,26]. İşte ham veriyi bilgiye veya anlamlı hale dönüştürme işini veri madenciliği ile yapabiliriz.

Peki Veri Madenciliğini bu anlamda önemli kılan nedir? Her şeyden önce Veri Madenciliğinde amaç, anlamsız veri yığınlarından veya işlenmemiş verilerden anlamlı ve kullanışlı bilgiyi çıkarmak ve bu bilgiyi işlemektir. Böylece, işlenmiş bilgilerden mevcut ham verilere göre daha çok yararlanılabilir.

1.1. Veri Madenciliği

Veri madenciliği, eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır. Bu da; kümeleme, veri özetleme, değişikliklerin analizi, sapmaların tespiti gibi belirli sayıda teknik yaklaşımları içerir [7].

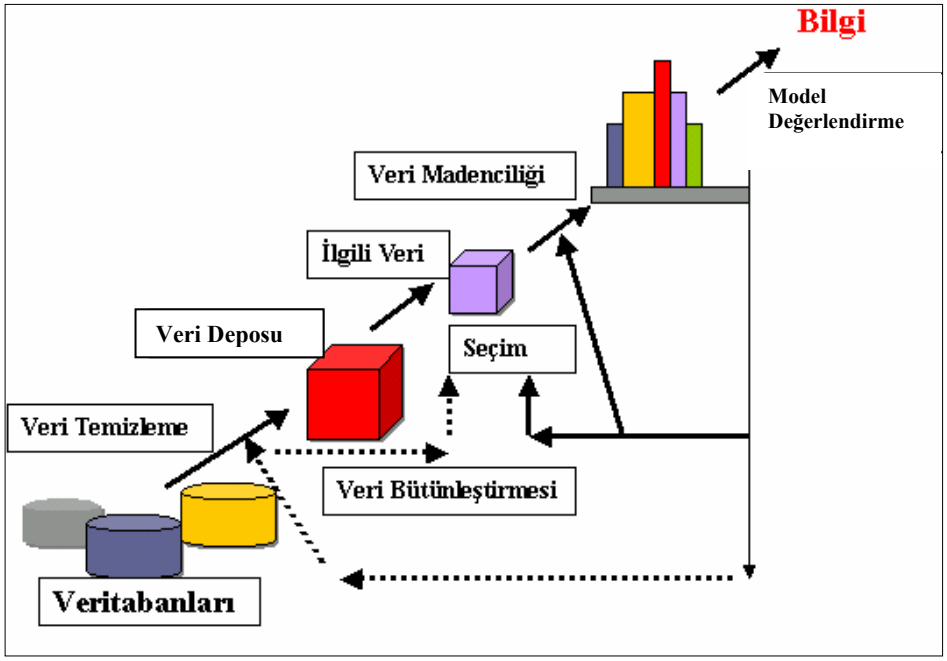
Başka bir deyişle, veri madenciliği, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir. Veri madenciliğini istatistiksel bir yöntemler serisi olarak

görmek mümkün olabilir. Ancak veri madenciliği, geleneksel istatistikten birkaç yönde farklılık gösterir [7].

Veri madenciliğinde amaç, kolaylıkla mantıksal kurallara ya da görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır. Veri madenciliğin de, kullanıcı ve bilgi tabanı etkileşim halindedir. Veriler arasındaki ilginç bağlantılar modellenir, kullanıcıya gösterilir ve istenirse bilgi, veri tabanına da kaydedilebilir. Buna göre, veri madenciliği işlemi, gizli kalmış veri bağlantıları bulunana kadar devam eder.

Veri madenciliği, çok büyük miktardaki ham verilerden önemli bilgilerin çıkarılmasıdır. Başka bir bakışla, anlamsız bilgi yığınlarından veya işlenmemiş verilerden anlamlı ve kullanışlı bilgiyi çıkarmak ve bu bilgiyi işlemektir. Veri madenciliği basit bir sorgu işlemi değildir. Uzman sistemler (*expert system*), öğrenen sistemler, istatistiksel ya da basit bir veri tabanı işlemi de değildir [20,26].

Veri madenciliğinin temelinde, veri kümeleri arasındaki ilişkilerin ve benzerliklerin bulunması, verinin analiz edilmesi, yazılım tekniklerinin ve istatistiksel yöntemlerin kullanılması vardır. Anlamsız verilerin işlenip anlamlı bilginin oluşturulmasına kadar, veriler birçok işlemden geçirilir [28]. Bu işlemler veri madenciliğinin aşamalarını oluştururlar. Bu aşamalar özet olarak Şekil 1. [23]' de gösterilmiştir.



Şekil 1. Veri madenciliği süreçleri

Şekil 1.' de görülen aşamalar bir veri tabanlarındaki ham verinin farklı disiplinler (Matematik, İstatistik, Bilgisayar Bilimleri, Biyoloji, Kimya, Tıp v.s.) yardımıyla amaca uygun olarak işlenmesi ve hedeflenen anlamlı bilginin elde edilmesini göstermektedir. Aslında veri madenciliği bilgi keşif sürecinin bir parçasıdır. Bilgi keşif sürecinde verinin, Şekil 1' de gösterilen analiz aşamaları ;

1-Veri Temizleme : Gürültülü ve tutarsız veriler veri tabanından temizlenir. Bilgi yığını, ön filtreleme işlemlerine tabi tutularak gereksiz verilerden arındırılır.

2-Veri Bütünleştirme : Bir çok veri kaynağı birleştirilir ve ortak alandaki veriler bütünleşik olarak işleme dahil edilirler.

3-Veri Seçme : Yapılacak olan analiz ile ilgili işe yarar verileri belirlemek ve bu verilerle sürece devam etmek.

4-Veri Dönüşümü : Verinin, veri madenciliği teknikleriyle kullanılabilecek biçime dönüşümünü gerçekleştirmek.

5-Veri Madenciliği : Veri ilişkilerini yakalayabilmek için akıllı metotları ve algoritmaları uygulamak.

6-Model Değerlendirme : Bazı ölçümlere göre elde edilmiş bilgiyi temsil eden ilginç bağlantıları modellemek ve tanımlamak.

7-Bilgi Sunumu : Modellenmiş ve tanımlanmış bilginin kullanıcıya sunumunu gerçekleştirmek.

Bu aşamaların sonucunda ham veri işlenir, veriler arası bağlantılar belirlenir, bu bağlantılar modellenir ve bu modelle bilgi keşfi yapılmış olur [28].

1.1.1. Veri Madenciliğinde Kullanılan Teknikler

1.1.1.1. İstatistiksel Yöntemler

Veri madenciliği çalışması esas olarak bir istatistik uygulamasıdır. Verilen bir örnek kümesine bir kestirici oturtmayı amaçlar. İstatistik literatüründe son elli yılda bu amaç için değişik teknikler önerilmiştir. Naive Bayes olasılık metodu, sınıflandırma da kullanılan istatistiksel yöntemlerin başında gelir. Bu teknikler istatistik literatüründe çok boyutlu analiz (*multivariate analysis*) başlığı altında toplanır ve genelde verinin parametrik bir modelden (çoğunlukla çok boyutlu bir Gauss dağılımından) geldiğini varsayar. Bu varsayım altında sınıflandırma (*classification; discriminant analysis*), regresyon, öbekleme

(*clustering*), boyut azaltma (*dimensionality reduction*), hipotez testi, varyans analizi, bağıntı (*association; dependency*) kurma için teknikler istatistikte uzun yıllardır kullanılmaktadır [4].

1.1.1.2. İlişkilendirme Kuralları (*Association Rules*)

Genellikle alışveriş uygulamalarında kullanıldığı için İlişkilendirme Kuralları aynı zamanda Alış Veriş Sepeti (Market Basket) analizi olarak da tanınmaktadır. Bu yöntemdeki amaç bir küme içerisindeki nesnelerin birbirleri ile olan bağlarının tespit edilmesidir. Bu Veri Madenciliği yöntemi yaygın olarak alışveriş sistemlerinde kullanıldığı görülse de başka uygulamalarda da kullanılmaktadır.

İlişkilendirme Kuralı yöntemine örnek verecek olursak A ürününün alınması ile B ürününün veya C ürünün alınması arasında bir bağlantı olup olmadığının tespit edilmesi ve eğer bağlantı var ise bu bağlantılar arasındaki kuvvet veya önem derecesinin (confidence or strength) ortaya çıkarılması sağlanır. Bu analizin amacı A ürününü alan kişilerin B veya C ürünleri alımlarıyla ilgili olarak kuvvetli bir bağlantı bulup sistemde bir takım değişiklikler gerçekleştirmektir. Örneğin, süper market sisteminde çeşitli promosyonların gerçekleşmesi, ürün raflarının elde edilen sonuçlar doğrultusunda yerleştirilmesi olabilir. Bu işlem, bir web sitesi içerisinde sayfaların yapılandırılmasında kullanılır [4].

1.1.1.3. Sıralı Patern (*Sequential Patern*)

Sıralı patern yöntemiyle kullanıcı oturumları arasında patern bulunmaya çalışılır. Sıralı patern bulma işleminde, belirli zaman aralıklarında oturumlar incelenir ve karşılaştırma yapılır. Sıralı patern yönteminde, eğilim analizi, değişen nokta bulma veya benzerlik analizleri gibi bazı geçici analiz tipleri kullanır. Sıralı paternlerin bulunması, örneğin gelecekteki eğilimi tahmin edecek web pazarlamacıları için oldukça anlamlıdır. Böylece ilanlar belirli kullanıcı gruplarına yönlendirilebilir.

1.1.1.4. Kümeleme (*Clustering*)

Kümeleme yöntemi aynı karakteristiğe sahip olan nesnelerin bir araya getirilmesi işlemidir. Web Madenciliğinde genel olarak iki kümeleme yaklaşımı vardır: Kullanıcı Grupları (*User Clusters*), Sayfa Grupları (*Page Clusters*).

1.1.1.5. Bellek Tabanlı Yöntemler

Bellek tabanlı veya örnek tabanlı bu yöntemler (*memory-based, instance-based methods; case-based reasoning*) istatistikte 1950'li yıllarda önerilmiş olmasına rağmen o yıllarda gerektirdiği hesaplama ve bellek yüzünden kullanılamamış ama günümüzde bilgisayarların ucuzlaması ve kapasitelerinin artmasıyla, özellikle de çok işlemcili sistemlerin yaygınlaşmasıyla, kullanılabilir olmuştur. Bu yönteme en iyi örnek en yakın *k* en yakın komşu algoritmasıdır (*k-nearest neighbor*) [4].

1.1.1.6. Yapay Sinir Ağları

1980'lerden sonra yaygınlaşan yapay sinir ağlarında (*artificial neural networks*) amaç fonksiyon birbirine bağlı basit işlemci ünitelerinden oluşan bir ağ üzerine dağıtılmıştır [4]. Yapay sinir ağlarında kullanılan öğrenme algoritmaları veriden üniteler arasındaki bağlantı ağırlıklarını hesaplar. YSA istatistiksel yöntemler gibi veri hakkında parametrik bir model varsaymaz yani uygulama alanı daha geniştir, ve bellek tabanlı yöntemler kadar yüksek işlem ve bellek gerektirmez.

1.1.1.7. Karar Ağaçları

İstatistiksel yöntemlerde veya yapay sinir ağlarında veriden bir fonksiyon öğrenildikten sonra bu fonksiyonun insanlar tarafından anlaşılabilecek bir kural olarak yorumlanması zordur. Karar ağaçları ise veriden oluşturulduktan sonra ağaç kökten yaprağa doğru inilerek kurallar (*IF-THEN rules*) yazılabilir [4]. Bu şekilde kural çıkarma (*rule extraction*), veri madenciliği çalışmasının sonucunun gerçeklenmesini sağlar. Bu kurallar uygulama konusunda uzman bir kişiye gösterilerek sonucun anlamlı olup olmadığı denetlenebilir. Sonradan başka bir teknik kullanılacak olsa bile, karar ağacı ile önce bir

kısa çalışma yapmak, önemli değişkenler ve yaklaşık kurallar konusunda bize bilgi verir ve tavsiye edilir.

1.1.2. Veri Madenciliğinin Kullanım Alanları

Veri madenciliği günümüzde yaygın olarak bir çok farklı alanda, farklı amaçlar için kullanılmaktadır. Aslında bilgisayarla entegre olan sektörler de daha yaygın olarak kullanılmaktadır [26]. Bu kullanım alanlarını, sektörlere gruplayacak olursak;

Bilişim ve Mühendislik

- Temel olarak verilerin sınıflandırmasın da,
- Görüntü işleme de,
- Doku sınıflandırma da,
- Yapay sinir ağları modellerinde,
- Dil işleme ve morfolojik analizlerde,
- Web kullanıcılarının modellenmesinde,
- Web sitelerinin güvenliğinde,
- Saldırı tespiti ve kimlik belirlemede,
- Web deki bilgilerin sınıflandırması ve arama sonuç modellemesinde,
- Ampirik veriler üzerinde modeller kurarak bilimsel ve teknik problemlerin çözümlenmesi.

Pazarlama

- Müşteri analizinde,
- Müşterilerin farklı özellikleri arasındaki bağlantıların kurulmasında,
- Farklı müşteri portföylerinin modellenmesinde
- Çeşitli pazarlama kampanyalarında,
- Mevcut müşterilerin elde tutulması için geliştirilecek pazarlama stratejilerinin oluşturulmasında,
- Market sepeti analizinde,
- Çapraz satış analizleri,
- Müşteri değerleme,
- Müşteri ilişkileri yönetiminde,
- Satış tahminlerinde,

- Kar tahminlerinde,
- Kriz yönetimlerinde,
- Yatırım stratejileri geliştirmede

Bankacılık

- Farklı finansal göstergeler arasındaki gizli korelasyonların bulunmasında,
- Kredi kartı dolandırıcılıklarının tespitinde,
- Müşteri analizinde,
- Kredi taleplerinin değerlendirilmesinde,
- Usulsüzlük tespiti,
- Risk analizleri,
- Risk yönetimi,
- Yatırımların yönetiminde,
- İnsan kaynakları yönetiminde,
- Müşteri kitlelerinin modellenmesinde

Sigortacılık

- Yeni poliçe talep edecek müşterilerin tahmin edilmesinde,
- Sigorta dolandırıcılıklarının tespitinde,
- Riskli müşteri tipinin belirlenmesinde,
- Perakendecilik,
- Satış noktası veri analizleri,
- Alış-veriş sepeti analizleri,
- Tedarik ve mağaza yerleşim optimizasyonu,

Borsa

- Hisse senedi fiyat tahmini,
- Genel piyasa analizleri,
- Alım-satım stratejilerinin optimizasyonu,
- Telekomünikasyon
- Kalite ve iyileştirme analizlerinde,
- Hisse tespitlerinde,
- Hatların yoğunluk tahminlerinde,
- Borsa yatırımcılarının modellenmesinde,

Sağlık ve İlaç

- Test sonuçlarının tahmini,
- Ürün geliştirme,
- Tıbbi teşhis,
- Tedavi sürecinin belirlenmesinde,
- Ürün verimliliğini ölçme de
- Hedef kitle belirleme de,

Endüstri

- Kalite kontrol analizlerinde
- Lojistik,
- Üretim süreçlerinin optimizasyonun da,
- Yön eylem araştırmalarında,
- Strateji belirlemede [26].

1.2. Web Madenciliği

World Wide Web büyük bir bilgi dağı olarak, her geçen gün daha da büyümekte ve güncel hayatımızda daha çok yer almaktadır. İnternetin herkesten bağımsız ve herkese açık olması, İnternetin ve içerdiği bilginin sürekli ve de düzensiz olarak hızla büyümesine neden olmaktadır. Bunun sonucunda İnternet, karşımıza, düzensiz ve sürekli büyüyen bir bilgi yığını olarak çıkmıştır.

Veri Madenciliğinde işlenen verilerin büyük bir kısmını web dokümanları oluşturmaktadır. Bu durum veri madenciliği içinde "Web Madenciliği" diye yeni bir alanın doğmasına yol açmıştır. İnternet ortamındaki verilerin büyük olması kadar bu bilgilerin düzensiz oluşu da bu noktada web madenciliğine ayrı bir önem kazandırmaktadır.

Web madenciliği ilk kez 1996 yılında Oren Etzioni tarafından dile getirilmiştir. Web madenciliği, veri madenciliği tekniklerinin kullanılarak web belgelerinden ve servislerinden otomatik olarak bilginin ayıklanması, ortaya çıkarılması ve tahlil edilmesidir [1,20]. Web madenciliği, veri madenciliği tekniklerinin Web üzerinde uygulanması anlamına gelmektedir. Burada kullanılan ham veri, web dokümanlarıdır.

Etzioni web madenciliğini 3 alt işleme ayırmıştır ama günümüzde daha çok Web madenciliği 4 alt işlem şeklinde gösterilir [1,20] :

1. Resource Finding (Kaynak bulma)
2. Information Extraction and Pre-Processing (Bilgi Çıkarımı ve Ön İşleme)
3. Generalization (Genelleştirme)
4. Analysis (Çözümleme)

Aslında kaynak bulma çok geniş kapsamlı bir konudur ama kısaca bilginin kazanımı, elde etme kısmı diyebiliriz. Bunu da çeşitli verileri çevrimiçi ya da çevrimdışı olmasına bakmadan bir veri ambarında toplayarak yapmaktır. Web deki çeşitli gazeteler, haber grupları, portal sayfaları gibi yerlerden verinin toplanıp tabiî ki HTML etiketlerinden ayrılıp (ham veri) arama amaçlı bir yerde saklanmasıdır. Burada önemli bir nokta da, bu yapılırken verinin indekslenmesi ve arandığında konu başlıklarına göre çok hızlıca bulunması sağlanmaktadır [20].

Günümüzde, Web Madenciliği, veri madenciliğinin en önemli alt dallarından biridir. Web in gösterdiği gelişime paralel olarak, Web Madenciliği çok hızlı gelişmiş ve kendi içinde alt dallara bölünmüştür. Bu alt dallar Web İçerik Madenciliği, Web Yapı Madenciliği ve Web kullanım Madenciliği adı altında 3 grupta toplanmıştır [20].

1.2.1. Web Yapı Madenciliği

Web erişim araçlarının çoğu çok değerli olabilecek link verisini göz ardı ederek sadece metin verisine ulaşır. Web yapı madenciliğinin amacı web sitesi ve web sayfası hakkında bağlantı verisine bakarak bilgi üretmektir. Teknik olarak, Web içerik madenciliği dokümanın içeriğine, yapı madenciliği ise dokümanlar arası bağlantılara yoğunlaşır. Web yapı madenciliği, linklerin topolojisine dayanarak farklı siteler arsındaki benzerlik ve ilişki gibi bilgileri üretir. Ayrıca web yapı madenciliği sayfaların link tasarımlarını ortaya çıkarmamıza yardımcı olur. Web sayfalarında bulunan site haritası veya bağlantı haritası bu bağlamda sıklıkla kullanılır.

Linkler aslında bize çok önemli bir hizmet sağlamaktadırlar. Web üzerinde iki nokta arasındaki en kısa yolu sağlamaktadırlar. Bu bizim için ve web sayfaları için son derece önemlidir. Eğer bir sayfa bir başka sayfaya doğrudan linklenmişse bu o iki sayfa arasındaki bağlantıyı ve komşuluğu gösteren en güzel örnektir. Aslında doğası gereği web içerik madenciliğine çok yakın bir konudur. Bir sayfaya gelen linklere bakarken, bunları ikiye

ayırmalıyız, sitenin içinden gelen linkler ve başka sitelerden gelen linkler. Ayrıca sayfaya gelen linkler “in-degree” , sayfadan başka düğümlere giden linklere ise “out-degree” denmektedir.

Burada sonuç olarak şunları söyleyebiliriz ki, bir sayfayla ilgili tüm sayfaları bulmakta bu yöntem kullanılmaktadır. İstenen konuyla ilgili bir tane sayfanın sisteme verilmesiyle onunla ilgili tüm sayfalara erişebiliriz [20].

1.2.2. Web Kullanım Madenciliği

Web kullanım madenciliği kullanıcın siteyi kullanırken gerisinde bıraktığı erişim verilerinden bilgi üretmeyi amaçlar. Bu veriler ikinci sınıf verilerdir yani bir yere girilmiş; bir yerde yazılan ya da kullanıcın isteğiyle oluşan, linke tıklamak gibi, veri değildir. Tamamen kullanıcıdan bağımsız oluşur ve çok ciddi boyutlardadır. Bu veriler istemcilerde, sunucularda ve Proxy sunucularında depolanır. Veri kaynakları olarak sunucu erişim kayıtları, referrer kayıtları, agent kayıtları, istemci tarafında bulunan çerezler (cookies), kullanıcı profilleri (kayıt bilgileri), metadata (sayfa özellikleri, içerik özellikleri, kullanılan veri) sayılabilir.

Bu çözümleme, bize müşteri değerlerinin yaşam sürelerini belirlememize yardımcı olacaktır. Web pazarlama uygulamalarında kampanyaların sonuçlarını anlamamıza yarayacak ve daha efektif bir yönetim oluşturmamızı sağlayacaktır. Kullanıcı erişim izlerinin çözümlenmesi sonucunda hedef müşteri kitleleri oluşturabiliriz. Web kullanım madenciliği sitemizdeki hedef kitleye yönelik stratejiler üretebilir, sitemizin hangi kitleler tarafından hangi amaçla kullanıldığını gözlemleyebiliriz [20].

1.2.3. Web İçerik Madenciliği

Web içerik madenciliği, web dokümanlarının belli istatistiksel metotlarla işlenip dokümanlardan istenen bilgilerin elde edilmesini sağlar. Böylelikle Web dokümanlarından yararlı ve gerekli bilgi elde edilir. Web üzerindeki verilerin farklı türlerde olması, metin, ses, görüntü, resim, tablo, yardımcı veri, bağ verisi gibi farklı tiplerdeki yapısız veri, Web içerik madenciliğinde verinin analizini zorlaştırır. Bu yüzden farklı türlerdeki verilerin daha kolay analiz edilebilmesi için farklı yaklaşımlar geliştirilmiştir.

Bu yaklaşımlar iki grupta toplanır [20]:

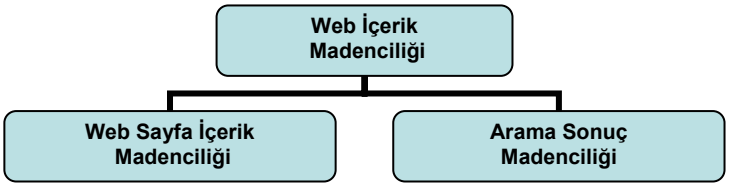
1) Bilgiye erişim.

2) Veri tabanı.

Bilgiye erişim yaklaşımları, genel olarak, bilgiye erişim, bilginin analizi ve ayrıştırılması, analiz edilen bilgilerin amaca uygun olarak sınıflandırılması konularında yoğunlaşır. Veri tabanı yaklaşımları, Web üzerinde bulunan verilerin bir veri tabanı üzerine kaydedilip, bu veri tabanı üzerinden modellenmesi, sorgulanması ve filtrelenmesi üzerinde çalışır.

Web içerik madenciliğinde kaynak veri, web sayfalarının içeriklerinden oluştuğu için kullanılan veri tamamen sınıfsız ve karmaşık bir veridir. Bu yüzden web sayfalarının bir veri tabanı üzerinde modellenmesi işleri daha kolay hale getirir. Sorgulama, filtreleme ve sınıflandırma işlemleri veri tabanı üzerinde daha kolay yapılır.

Web içerik madenciliği Şekil 2'de verildiği gibi işlenen verinin türüne göre iki gruba ayrılır.



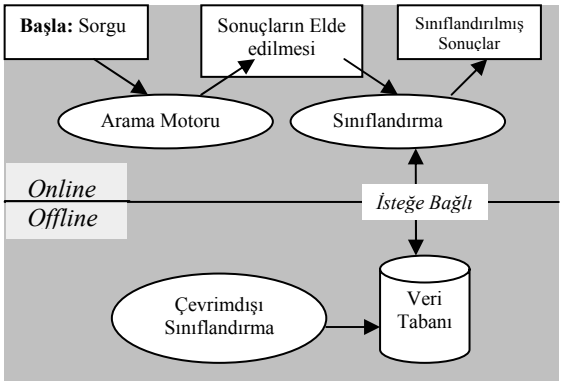
Şekil 2. Web içerik madenciliği sınıflandırması

Web sayfa içerik madenciliği, web sayfalarının içeriklerinden farklı bilgiler elde etmek için kullanılır. Bu bilgi metin olabileceği gibi sayfalardan resim, video veya ses gibi değişik formatlardaki bilgilerinde çıkarımı sağlanabilir.

Arama sonuç madenciliği ise, arama motorlarından elde edilen arama sonuçlarının sınıflandırılmasını sağlar. Bu sayede arama sonuçlarından, aranan hedefe daha yakın olanlarının saptanması mümkün olabilmektedir. Bu çalışmada arama sonuç madenciliği teknikleriyle, arama sonuçları analiz edilerek hangi konu ile ilgili olduğu tahmin edilmeye çalışılmıştır.

1.2.3.1. Arama Sonuç Madenciliği

Arama sonuç madenciliği, arama motorlarından ede edilen sonuçların sınıflandırılması ile ilgilidir. Bir aramadan elde edilen sonuçların gerçekte bizim aradığımız konuya ne kadar yakın olduğu araştırılır. Bir arama motorunda bir arama sonucunda yüzlerce sonuç üretilir. Fakat, bu yüzlerce sonucun içinde ancak sınırlı sayıdaki sonuç gerçekte bizim aradığımız sayfalardır. Sistemi biraz daha açacak olursak, sistem iki türlü çalışabilmektedir. Çevrimiçi (online) olarak sayfalara bağlanıp veri analizi yapılmaktadır. Bunun yanında isteğe bağlı olarak sayfalar bir veri tabanına kaydedilip internet ten bağımsız olarak veri tabanı sorgulama araçlarıyla analiz edilebilmektedir. Sistemin genel yapısı ve çalışma şekli, Şekil 3. teki blok diyagram da detaylı olarak verilmiştir.



Şekil 3. Gerçeklenen sistemin aşamaları

İnternetteki büyümenin giderek hızlanması ve veri miktarının sürekli artması, aranan doğru bilgiye erişim sürecini hiç şüphesiz olumsuz etkilemektedir. Bu durumda web deki veri artışına paralel olarak arama motorları üzerinde ikinci hatta daha yüksek derecelerde filtreleme yapılarak doğru bilgiye erişim ancak sağlanabilecektir. Bu da tahminlerin üzerinde bir zaman kaybı demektir. Birde aranan sayfalar analiz edildiğinde hiç de bizim istemediğimiz içeriklere rastlıyorsak bu sayfaları boşuna taramış olacağız ve zaman kaybı daha da artacaktır.

Arama sonuç madenciliği, bu yüzden çok önemli bir kavram olarak karşımıza çıkıyor. Üzerinde çalıştığımız sistem, gerekli filtrelemeleri yaparak, doğru bilgiye daha kısada sürede nasıl erişebilir, sorusunu çözmeye yöneliktir.

Sadece kelimelerin istatistiksel değerlendirilmelerine dayanan içerik madenciliği bazı durumlarda başarısız olabilmektedir. Örneğin, seçilen kelimelerin aranan sayfada çok az sayıda bulunması durumunda başarısız sonuçlar üretilebilmektedir. Burada, anlamsal içerik analizi, konu madenciliğinde başarının artırılmasına önemli katkılar sağlayabilir. Bu durumda, seçilen kelime ve kelime gruplarına dayanan istatistiksel sonuçlarla, anlamsal analiz sonuçları ile birlikte değerlendirilerek daha doğru sonuçlara ulaşılabilir.

Bu çalışmada, Web Madenciliğinin alt dallarından Web İçerik Madenciliği ele alınmış ve içerik madenciliği için elde edilen arama sonuçlarından yeni bilgilerin çıkartımı ve bilgilerin sınıflandırılması amaçlanmıştır.

Günümüzde, web ortamında bilgiye ulaşmak için birçok arama motorundan yararlanılmaktadır. Bu arama motorları, bilginin ilişkilendirilmesi ve sınıflandırılması yönünden oldukça sınırlı ve kullanışsızdırlar. Dolayısıyla, web ortamında elde edilen bilgilerin sınıflandırılması ve ilişkilendirilmesi konusunda halen birçok araştırma ve çalışma sürdürülmektedir.

1.3. Veri Sınıflandırma Algoritmaları

Veri madenciliğinde, verilerin sınıflandırılmasında birçok yöntem ve algoritma kullanılmaktadır. Başlıca sınıflandırma algoritmaları;

- Karar ağaçları,
- SVM (Vektör Destek Makinası) Algoritması,
- Yapay Sinir Ağları
- Naive Bayes Olasılık Metodu,
- K-NN (K- En yakın Komşu) Algoritması,
- Genetik Algoritmalar,
- Regresyon Analizleri,

algoritmalarıdır. Bu algoritmalar veri sınıflandırma ve analizlerde iç içe veya bağımsız olarak kullanılırlar. Bu algoritmalardan Karar Ağaçları, Naive Bayes Olasılılık Metodu, K-NN algoritması ve SVM algoritması yapılan çalışma kapsamında web metinleri üzerinde uygulanmış ve metin sınıflandırma performansları karşılaştırılmıştır. O yüzden bundan sonraki bölümde bu 4 algoritmanın çalışma yapısı ve modellenmesi daha detaylı olarak açıklanmıştır.

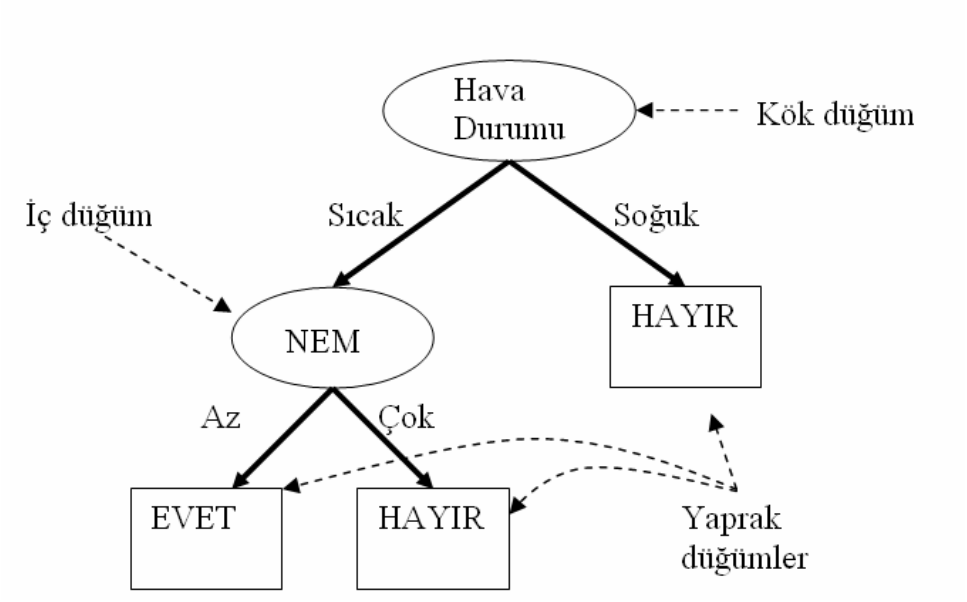
1.3.1. Karar Ağaçları

Veri madenciliğinde bir karar ağacı, veriyi değil kararları temsil eder. Karar ağaçları, bir ağaç grafiği veya bir kararlar modeli olarak, olası sonuçları yorumlamayı sağlayan bir karar destek aracıdır. Karar ağaçları, sıklıkla yöneylem araştırmalarında, karar analizi ve özellikle bir strateji de en önemli hedefe giden yolu olasılıklarla belirlemek için kullanılır. Karar ağaçları ayrıca, koşullu olasılık hesaplarında açıklayıcı bir model olarak kullanılır. Karar Ağaçları, veri madenciliği ve makina öğrenmesinde kullanılan en temel yöntemlerin başında gelmektedir. Bunun başlıca sebepleri, anlaşılması ve yorumlanmasının oldukça basit olması, diğer algoritmalarla kolay entegre olması, daha zahmetsiz ve uygulanabilir olmasıdır. Veri madenciliği ve makina öğrenmesinde bir karar ağacı, akıllı bir model olarak kullanılır. Hedef değer ile ilgili sonuçlar gözlemlerle eşleştirilir. Karar ağaçları için iki model vardır. Sınıflandırma ağacı (kesikli sonuç) veya Regresyon ağacı (sürekli sonuç). Karar ağaçları düğümler ve dallardan oluşan , anlaşılması oldukça kolay olan bir tekniktir. En basit şekilde her dal kendinden sonra evet veya hayır şeklinde en az iki dala ayrılmaktadır. Karar ağacında bulunan her bir dalın belirli bir olasılığı mevcuttur. Bu sayede son dallardan köke veya istediğimiz yere ulaşana dek olasılıklar birbiriyle çarpılarak hedefe giden en yüksek olasılıklı yol, hesaplanmış olur. Hesaplamanın verimliliği ağacın belirli dalları kesilerek yada belirleyici özellikler değiştirilerek, yani daha az yararlı kurallardan kurtularak geliştirilebilir. Daha önceden de söylediğimiz gibi karar ağaçları anlaşılması oldukça kolay olduğundan dolayı sıklıkla başvurulan bir yöntemdir. Genellikle, yöneylem araştırmalarında büyük veri yığınlarının sınıflandırılması için kullanılır [22,23].

Ağacın 3 tür düğümü vardır:

- Kendisinden önce bir dal olmayan ve kendisinden sıfır veya daha fazla dal çıkabilen Kök düğümü
- Kendisinden önce ona doğru gelen sadece bir dal olan ve kendisinden iki veya daha fazla dal çıkan İç düğümler
- Kendisinden önce ona doğru gelen sadece bir dal olan ve kendisinden hiç dal çıkmayan Yaprak veya kutup (terminal) düğümler

Aşağıdaki şekilde hava durumu futbol oynamaya uygun mu sorusunun karar ağacı ile çözümü yer almaktadır.



Şekil 4. Hava durumu problemi için bir karar ağacı

Karar ağacı kullanarak sınıflandırma;

Avantajları

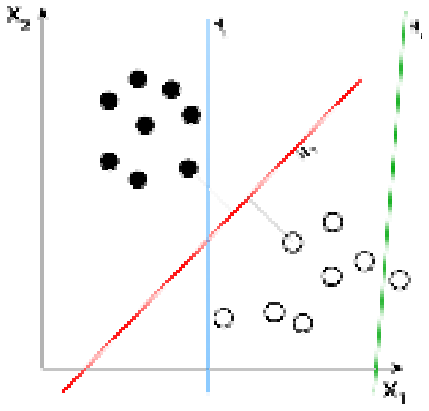
- Karar ağacını anlamak ve yorumlamak kolaydır.
- Karar ağaçları, hazırlık için az veri gerektirir.
- İstatistiksel testleri doğrulamak için güçlü bir modeldir.
- Karar ağacı oluşturmak zahmetsizdir.
- Hesaplama gerektirmeden sınıflandırma yapabilir.
- Sürekli ve kesikli nitelik değerleri için kullanılabilir.
- Büyük verileri kısa sürede modellemek mümkündür.
- Diğer algoritmalara entegre edilebilir.

Dezavantajları

- Sürekli nitelik değerlerini tahmin etmekte çok başarılı değildir.
- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma zorlaşır.
- Zaman ve yer karmaşıklığı öğrenme kümesi örnekleri sayısına (q), nitelik sayısına (h) ve oluşan ağacın yapısına bağlı.
- Ağaç oluşturmak için zaman karmaşıklığı
- Ağaç oluşturma ve ağaç budama maliyetleri pahalı olabilir [22,23].

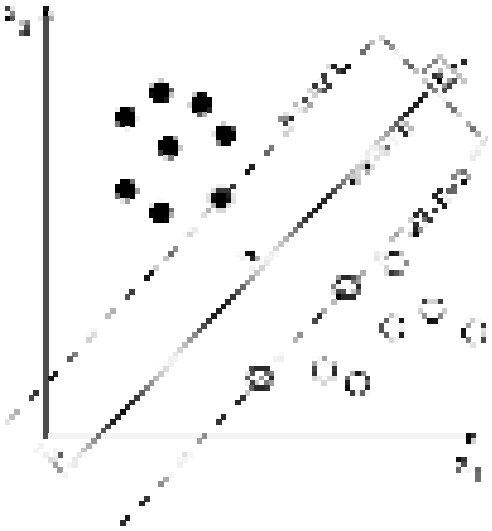
1.3.2. SVM (Vektör Destek Makinası) Algoritması

Vektör Destek Makinası Algoritması, sınıflandırmada kullanılan, bir denetimli öğrenme algoritmasıdır. Bu algoritma son 10 yıl içinde geliştirilmiştir. Sınıflandırma konusunda kullanılan oldukça etkili ve basit yöntemlerden birisidir. Bir harita düşünelim. Haritadaki iki bölgeyi en iyi şekilde birbirinden ayıracak bir sınır çizgisi çekmek istiyoruz. En doğru çizgiyi çekmeliyiz ki bölgeleri doğru şekilde ayıralım. SVM algoritmasının temelinde bu problemin çözünü vardır. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır. İşte SVM bu sınırın nasıl çizileceğini belirler. Farz edelim ki pozitif ve negatif örnekleri birbirinden ayıran bir aşırı düzlem var, bu düzlem üzerindeki noktalar w.x+b=0 eşitliğini sağlayacaktır, burada w aşırı düzleme olan normal ve |b|/||w|| aşırı düzlemden orijine olan dik uzaklıktır. Aşırı düzleme en yakın pozitif ve negatif örnekler arasındaki mesafeye ayırıcı aşırı düzleminin "tolerans" ı dersek, destek vektör yöntemi bu "tolerans"ın en yüksek olduğu bir aşırı düzlemi bulmaya çalışır.



Şekil 5. Toleransın belirlenmesi

Bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu sınır çizgileri birbirine yaklaştırılarak ortak sınır çizgisi üretilir. Örneğin Şekil 6. da gösterilen iki grubu ele alalım:



Şekil 6. Grupların düzlemde gösterilmesi

Bu şekilde iki grup iki boyutlu bir düzlem üzerinde gösterilmiştir. Bu düzlemi ve boyutları birer özellik olarak düşünmek mümkündür. Yani basit anlamda sisteme giren her girdinin (input) bir özellik çıkarımı (feature extraction) yapılmış ve sonuçta bu iki boyutlu düzlemde her girdiyi gösteren farklı bir nokta elde edilmiştir. Bu noktaların sınıflandırılması demek, çıkarılmış olan özelliklere göre girdilerin sınıflandırılması demektir [24,29].

Yukarıda her iki sınıf arasında oluşan aralığa tolerans (offset) demek mümkündür. Bu düzlemdeki her bir noktanın tanımı aşağıdaki gösterim ile yapılabilir:

$$\mathcal{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^{n} \qquad (1)$$

Yukarıdaki gösterimi şu şekilde okumak mümkündür. Her x,c ikilisi için X, vektör uzayımızdaki bir nokta ve c ise bu noktanın -1 veya +1 olduğunu gösteren değeridir. Bu noktalar kümesi i = 1 ‘den n’e kadar gitmektedir.

Yani bu gösterim bir önceki şekilde olan noktaları ifade etmektedir. Bu gösterimin bir aşırı düzlem (hyperplane) üzerinde olduğunu düşünürsek. Bu gösterimdeki her noktanın: $wx - b = 0$ denklemi ile ifade edilmesi mümkündür.

Buradaki w aşırı düzleme dik olan normal vektörü ve x noktanın değişen parametresi ve b ise kayma oranıdır. Bu denklemi klasik ax+b doğru denklemine benzetmek mümkündür.

Yine yukarıdaki denkleme göre b/||w|| değeri bize iki grup arasındaki mesafe farkını verir. Bu mesafe farkına daha önce tolerans (offset) ismini de vermiştik. Bu mesafe farkı denklemine göre mesafeyi en yüksek değere çıkarmak için yukarıdaki ilk şekilde gösterilen 0, -1 ve +1 değerlerine sahip 3 doğruyu veren denklemde 2/||w|| formülü kullanılmıştır. Yani doğrular arası mesafe 2 birim olarak belirlenmiştir.

Bu denkleme göre elde edilen iki doğru denklemi:

$$wx - b = -1 \quad (2)$$

$$wx + b = 1 \quad (3)$$

olarak bulunmuştur. Aslında bu denklemler doğruların kaydırılması sonucunda elde edilen en yüksek değerlerin bulunması işleminin bir sonucudur. Aynı zamanda bu denklemlerle problemin doğrusal ayrılabilir (linearly seperable) olduğu da kabul edilmiş olur.

Tahmin edileceği üzere iki grup arasındaki aşırıdüzlemin (hyperplane) tek yönlü olması mümkün değildir.

Yukarıdaki şekilde (Şekil 6.) iki farklı aşırı düzlem olasılığı bulunmasına karşılık SVM yönteminde bu olasılıklardan en büyük toleransa (offset) sahip olanı alınır. Sezgisel olarak, iyi bir ayrıştırmanın yapılabilmesi için, her iki sınıfa ait veri noktalarının, sınır çizgisine maksimum mesafede olması gerekir [24,29].

Avantajları

- Uygulaması kolay bir algoritmadır.
- Ayrıştırmanın iyi yapıldığı durumlarda, başarı oranı yüksektir.

Dezavantajları

- Doğrusal ayrışımın yapılamadığı durumlarda başarı oranı düşer.
- Çoklu sınıflandırma performansı düşüktür.
- Yakın sınıflar için tolerans belirlemek zordur.

1.3.3. Naive Bayes Olasılık Metodu

Naive Bayes sınıflandırma algoritması, saf bayes olasılık kuralına dayanır. Eldeki verilere göre hipotezlerin doğru olma olasılığına göre hareket eder. Gelen verilere göre maksimum olasılığa sahip hipotez seçilir. Niteliklerin hepsi aynı derecede önemli ve birbirinden bağımsızdır.

Naive Bayes sınıflandırıcı algoritması metin dokümanlarını sınıflandırmak için kullanılan en başarılı algoritmalardan biridir. Uygulanabilirliği kolay bir algoritma olup performans ve zaman bakımından da başarılıdır. Naive Bayes'e göre kelimeler sınıftan bağımsızdır [18].

Naive Bayes sınıflandırıcılar, bilinen bir sınıf için terim olasılıklarının hesaplanma yöntemine göre *çok terimli (multinominal)* ve *çok değişkenli (multivariate)* olmak üzere ikiye ayrılırlar. Çok terimli yöntemde terimlerin geçiş sayıları da dikkate alınırken, çok değişkenli yöntemde terimlerin sadece var olup olmadıklarına bakılır [5,6].

1.3.3.1. Naive Bayes Binary Ağırlıklandırma *(Multi-Variate Model)*

Klasik saf Bayes modeli, koşullu bağımsızlık ilkesine dayanır. Bu algoritma ile dokümanlarda kelimelerin olup olmamasına bakılarak ağırlıklandırma yapılır. Kelimelerin tekrar sayıları dikkate alınmaz. Dokümana ait kelime vektörü 1 ve 0' lardan oluşur. Kelime varsa 1, yoksa 0 kodlanır. Genellikle 2 olasılıklı durumlarda bu model tercih edilir. Örneğin aranan sayfa, sporla ilgilidir veya değildir.

Herhangi bir vektörün herhangi bir sınıfta olma olasılığını, bu modele göre hesaplamak için aşağıdaki eşitlikler kullanılır [5,6,18].

d vektörünün c_j kategorisinde olma olasılığı:

$$p(d|c_j) = \prod_{t=1}^{|V|} p(w_t|c_j)^{x_t}(1-p(w_t|c_j))^{(1-x_t)} \tag{4}$$

$$p(w_t|c_j) = \frac{1+B_{jt}}{2+|c_j|} \tag{5}$$

|V| : Sözlükteki kelime sayısı

Bjt : "cj" sınıfında bulunan ve "wt" kelimesini içeren eğitim dokümanı sayısı

|Cj| : "cj" sınıfında bulunan eğitim dokümanı sayısı

xt : Kelimenin ağırlığı (1 veya 0)

$$M(C) = P(word1|C)^{n1} P(word2|C)^{n2} ... P(wordv|C)^{nv} P(|C) \quad (6)$$

Dokümanın ait olduğu sınıfı belirlemek için, her sınıf için M(C) değeri hesaplanır. Doküman, M(C) değeri en büyük olan sınıfa atanır [5].

1.3.3.2. Naive Bayes Frekans Ağırlıklandırma *(Multinominal Model)*

Yapılan ölçümlere göre, doküman içindeki kelimelerin tekrar sayılarını hesaplarımızda kullanmanın, naive bayes binary modeline göre daha iyi çalıştığı görülmüştür. Elde edilen sınıflandırma başarısı binary modele göre oldukça artış göstermiştir. Bu artışla ilgili sayısal ölçümler sonuçlar bölümünde detaylı olarak verilmiştir. Multinominal modelde, her bir kelimenin tekrar etme sayısı diğer kelimelerin tekrar etme sayılarından bağımsızdır. Her kelime tekrar sayısı ile özellik vektöründe temsil edilir. Her dokümanda anahtar kelimelerin tekrar sayısı hesaplanır ve özellik vektörü bu sayılardan oluşur. Vektörün boyutu sözlükteki anahtar kelime sayısı kadardır. Yapılan çalışmada 300 anahtar kelimeden oluşan sözlük kullanıldığı için vektör boyutumuzda 300 olacaktır. Örnek olarak V= (2,5,0,0,4,0,12,0,1,……) şeklinde, boyutu 300 olan, farklı vektörler elde edilir [5].

d vektörünün c_j kategorisinde olma olasılığı:

$$p(d|c_j) = p(|d|)|d|! \prod_{t=1}^{|V|} \frac{p(w_t|c_j)^{xt}}{x_t!} \quad (7)$$

$$p(w_t|c_j) = \frac{1 + N_{jt}}{|V| + N_j} \quad (8)$$

d : Kategori Sayısı

N_{jt} : j sınıfındaki dokümanlar içinde t kelimesinin görülme sıklığı)

N_j : j sınıfındaki toplam kelime sayısı

P(|d|): Kategori olasılığı

x_t : Kelimenin frekansı

|V| : Kelime sayısı [5].

$$M(C) = P(word1|C)^{n1} P(word2|C)^{n2} ... P(wordv|C)^{nv} P(|C) \quad (9)$$

Dokümanın ait olduğu sınıfı belirlemek için, her sınıf için M(C) değeri hesaplanır. Doküman, M(C) değeri en büyük olan sınıfa atanır [5].

Avantajları

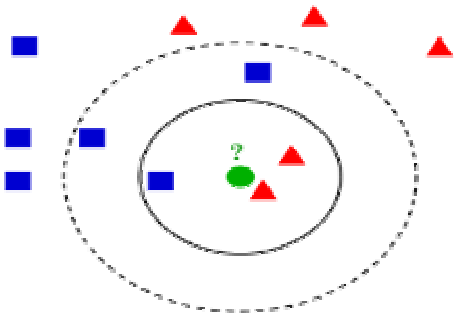
- Gerçeklemesi anlaşılır ve kolaydır.
- Çoğu durumda iyi sonuçlar verir.

Dezavantajları

- Varsayım: sınıf bilgisi verildiğinde nitelikler bağımsızdır.
- Gerçek hayatta değişkenler birbirine bağımlıdır.
- Değişkenler arası ilişki modellenemiyor. Değişkenlerin birbirinden bağımsız olması [23].

1.3.4. K-NN Algoritması

K en yakın komşuluk algoritması, tüm makine öğrenme algoritmaları arasında en basit, denetimli öğrenme algoritmasıdır. Algoritmanın eğitim aşamasında sınıf özelliği daha önceden belirlenmiş verilerden faydalanılır. K en yakın komşuluk algoritmasını uygulayabilmek için, metinlerin birer vektör olarak gösterilmesi gerekir. Aşağıdaki şekle göre yeşil daire sınıflandırılmak istenmektedir. Burada komşuluk için k parametresine ihtiyaç duyulur. K parametresi komşuluk sayısını belirtir. Birinci daire seçilirse k=3 seçilmiş olur. Çünkü birinci dairede 3 tane belirli sınıf var. İki tane kırmızı üçgen ve bir mavi kare. Bu durum için yani k=3 komşuluğunda kırmızı üçgen fazla olduğundan vektör, üçgen sınıfına atanır. Kesik çizgili daire alınırsa yani k=5 için, 3 kare 2 üçgen olduğundan vektör kare sınıfına atanır. Seçilen k komşuluğunda en fazla sınıflandırılmış veri hangi sınıftaysa, sorgu vektörü de o sınıfa atanır [30].



Şekil 7. K-en yakın komşuluğun tespiti

K en yakın komşuluk algoritmasını uygulayabilmek için, metinlerin birer vektör olarak gösterilmesi gerektiğini söylemiştik. Bu algoritmanın temelinde vektör-uzay modelinden faydalanılır. Metinler arasındaki benzerlikleri bulmak için, her metin vektörünün sorgu vektörü ile arasındaki açının kosinüs değeri hesaplanır. Vektör-uzay modelinde de bahsedildiği gibi sorgu vektörü, kosinüs değeri en yüksek olan vektörün sınıfına dahil edilir. İki vektör arasındaki açı ne kadar büyükse vektörler arasındaki benzerlik o derece az olacaktır. Açı küçüldükçe kosinüs değeri artacağından, kosinüs değeri 1'e en yakın olan metin vektörünün sınıfı, sorgu vektörünün atanacağı sınıftır. İki vektörün aynı olması durumunda aradaki açı sıfır olacak ve tam benzerlik olduğu anlaşılacaktır [5,6,30].

n boyutlu V_1 ve V_2 vektörleri arasındaki uzaklık, vektörlerin arasındaki açının kosinüsünün değerinin, aşağıdaki eşitlik yardımıyla hesaplanır.

$$\cos(\vec{v_1}, \vec{v_2}) = \frac{\sum_{i=1}^{n} v_{1i} v_{2i}}{\sqrt{\sum_{i=1}^{n} v_{1i}^{\,2}} \sqrt{\sum_{i=1}^{n} v_{2i}^{\,2}}} \qquad (10)$$

Kosinüs benzerliğini, metin vektörlerine uygulayacak olursak, d1 ve d2 iki metin dokümanına ait vektörler olsun. d1 ve d2 arasındaki Kosinüs benzerliği ;

$$cos(d1,d2) = d1*d2 / ||d1|| * ||d2|| \qquad (11)$$

di*dj: iki dokümanın vektör çarpımı.

||di||: di dokümanının uzunluğu.

Örneğin, *gol, hakem, taraftar, takım* anahtar kelimelerinden oluşmuş üç belgelik bir sistemde belgelerimiz aşağıdaki gibi olsun:

B_1: "gol, gol, gol, takım"

B_2: "hakem, hakem, hakem, hakem"

Q: "gol,hakem,taraftar,gol "

Bu durumda bu belgelere ilişkin vektörler şu şekilde olacaktır.

d_1: (3,0,0,1)

d_2: (0,4,0,0)

q : (2,0,1,0)

Normalizasyon ve ağırlıklandırmayı basitleştirme amacı ile yok sayarsak belgeler arası benzerlikler şu şekilde olacaktır.

sim(B_1,B_2)=0 ve sim(B_2,Q)=0

sim(B_1,Q)=0,848

Buna göre *sim(B_1,B_2)=0* olduğu için *B_1* ve *B_2* belgeleri arasında benzerlik yoktur. *Q* belgesi ile *B_1* arasında *0,848* oranında benzerlik vardır. Sınıflandırma yapılacak olsaydı bu sonuçla *Q* belgesi, *B_1* belgesinin sınıfına atanırdı.

Avantajları

- Uygulanabilirliği basit bir algoritmadır.
- Gürültülü eğitim dokümanlarına karsı dirençlidir.
- Eğitim dokümanları sayısı arttıkça verim artar.

Dezavantajları

- K parametreye ihtiyaç duyar.
- Uzaklık bazlı öğrenme algoritması, en iyi sonuçları elde etmek için, hangi uzaklık tipinin ve hangi niteliğin kullanılacağı konusunda açık değildir.
- Hesaplama maliyeti gerçekten çok yüksektir çünkü her bir sorgu örneğinin tüm eğitim örneklerine olan uzaklığını hesaplamak gerekmektedir. Bazı indeksleme metotları ile, bu maliyet azaltılabilir [5].

2. YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME

2.1. Ön İşlem Aşamaları

2.1.1. Metinlerin Çözümlenmesi

İnternet ortamında bulunan metinler, çok farklı biçimlerde olabilmektedir. Metinleri oluşturan kelimeler farklı ekler alarak, değişik anlamlarda kullanılmaktadır. Bu yüzden İnternet te bulunan bir metni olduğu gibi alarak sınıflandırma işlemine tabi tutamayız. Metin sınıflandırma işleminin doğru yapılabilmesi için kelimelerin eklerden ve noktalama işaretlerinden arındırılmış olması gerekir. Metinlerin bir çözümleme işlemine tabi tutulduktan sonra sınıflandırılması, sınıflandırmanın başarı yüzdesini doğrudan artırmaktadır. Türkçe dilinin yapısı gereği ön işleme zor olmaktadır. Çünkü Türkçe eklemeli bir dildir ve bir kelimeye eklenen her bir ek o kelimenin anlamını değiştirmektedir. Ayrıca eşanlamlı kelimelerden dolayı bir kelime farklı metinlerde, farklı anlamlar içerebilir. Türkçe bu karmaşık yapısıyla, İngilizce'den ve benzeri dillerden daha farklı metin çözümleme teknikleri gerektirir. Çözümleme sayesinde metinler küçük harfe dönüştürülür ve noktalama işaretlerinden arındırılır. Eklerle türetilmiş kelimeler için anahtar kelimelerle birlikte, joker kelimeler de kullanılacaktır. Yönetim programı anlatılırken çözümleme işleminden daha detaylı bahsedilecektir.

2.1.2. Kelime Vektörlerinin Oluşturulması

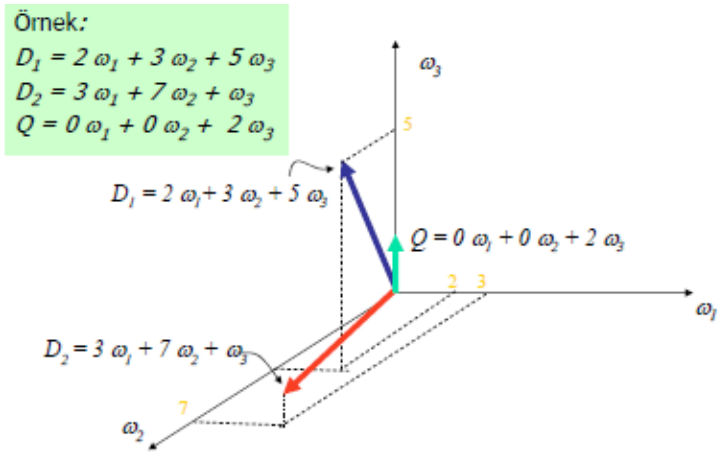
Bilindiği gibi metinler kelimelerden oluşur. Kelimeler statik yani sayısal değeri olmayan verilerdir. Sayısal olmayan veriler üzerinde de hiçbir matematiksel işlem, analiz veya sonuç çıkarma işlemi yapılamaz. Bu nedenle metinlerin sayılaştırılması veya sayı tablolarına dönüştürülmesi gerekir. Metinler ilk önce çözümleme işleminden geçirildikten sonra üzerinde herhangi bir işlemin yapılabilmesi için sayısallaştırılması gerekir. Bu işlem daha önceden belirlenmiş anahtar kelimelerin, metin içinde aranmasıyla elde edilen değerlerin frekansları ile oluşturulur. Kelime sayıları tespit edildikten sonra metin sayısal

olarak ifade edilebilir hale gelmiştir. Bu aşamadan sonra metin üzerine her türlü matematiksel ve istatistiksel veri analizi işlemlerini uygulayabiliriz.

Metinler üzerinde kategori işlemlerinin yapılabilmesi için daha önceden kategorilerin ve kelime vektörlerinin belirlenmesi gerekir. Bunun için, ilk önce metinlerde arayacağımız anahtar kelimelerin sözlüğünü oluştururuz. Kelime vektörümüzün boyutu daha önceden oluşturduğumuz sözlüğümüzün boyutuna eşit olacaktır. Sınıflandırma işlemine tabi tutulan her metin, uzayda temsili bir vektör ile gösterilir. Bu vektör, metin içinde geçen anahtar kelimelerin sayısını gösteren bir vektördür ve vektör boyutu sözlükteki kelime sayısına eşittir. Bu çalışma da kategorileri belirlemek için 300 anahtar kelimeden oluşan bir sözlük kullanılmıştır. Bu demektir ki, metinleri temsil eden her vektörün boyutu 300 olacaktır.

2.1.3. Vektör Uzay Modeli

Dokümanlar şekilde görüldüğü gibi kelimelerin vektörleri olarak ifade edilirler. ω'ler kelimelerin frekanslarını ifade etmektedirler [5,19]. D_1 *ve* D_2 ler eğitim dokümanlarının vektörünü, Q ise sınıflandırılacak dokümanın vektörünü göstermektedir. Burada sınıflandırılmak istenen dokümana ait Q vektörü, D_1 *ve* D_2 vektörleri ile karşılaştırılır. Vektörler arasındaki gerçek açıların hesaplanması yerine, vektörler arasındaki açının kosinüsü hesaplanır ve karşılaştırılır. Kosinüs değeri hangisinde yüksekse Q vektörü o dokümanın sınıfına atanır. Örneğin Q vektörü ile D_1 vektörü arasındaki açının kosinüsü, Q vektörü ile D_2 vektörü arasındaki açının kosinüsünden daha büyükse "Q vektörü, D_1 vektörü ile aynı kategoridedir" diyebiliriz.



Şekil 8. Vektör uzay modeli

Birçok metin sınıflandırma algoritması bu temele dayanır. Metin sınıflandırma işlemi için ilk yapılması gereken, sınıflandırılmak istenen dokümanlar ve daha önceden bir sınıfa atanmış olan eğitim dokümanları, vektör uzayında bir vektör olarak ifade edilmelidir [5].

2.1.4. Vektörlerin Ağırlıklandırılması

Bir metin sınıflandırılmadan önce çözümleme işleminden geçirilerek noktalama işaretlerinden arındırılır ve küçük harfe dönüştürülür. Sözlükteki kelimelerin sıklık sayısına göre, dokümanın kelime vektörü oluşturulur. Sözlükte kullanılan kelimeler, kelimenin kökünü içermektedir. Örneğin "enformatik" kelimesini arayalım. Eğer kelime, metinde "enformatiği" şeklinde geçiyorsa kelimeyi bulamaz. Bu için sonunda sert sessiz olan kelimeler için sonda bulunan sert sessiz harf düşürülerek sözlüğe eklenir. Yani "enformatik" kelimesi, "enformati" olarak sözlüğe eklenir. Bunun yanında programımızın çözümleme modülü, metni işaretlerden arındırdıktan sonra her kelimenin başına bir boşluk eklemektedir. Bu sayede metin içinde arama yaparken, her kelimenin ilk harfinden itibaren arama yapılır. Böyle olmasaydı "enformatik" kelimesi içinde enfo, format, forma, matik kelimelerini de bulurdu. Bunun sonucunda da tüm kelime frekansları yanlış hesaplanacaktı.

Bunu engellemek için programımızın çözümleme modülü içinde, metin temizlendikten sonra, her kelimenin başına bir boşluk karakteri ekleyerek aramanın her kelimenin başındaki boşluktan başlaması sağlanmıştır. Kelime içlerinde de boşluk karakteri olamayacağı için arama her zaman kelimenin başından yapılır ve kelime içindeki küçük kelime öbekleri arama dışında tutulur. Bu sayede, kelimelerin sayısı doğru şekilde hesaplanabilir. Bu çözümleme işleminden sonra, metin bir diziye aktarılır ve sözlükte bulunan kelimeler metin içinde aranır.

Örneğin , "Ekonomik krizden dolayı Hükümet , IMF ye olan kredi borçlarını ve borcun ödemesini yeniden görüşecek." metni çözümlendikten sonra "ekonomik krizden dolayı hükümet imf ye olan kredi borçlarını ve borcun ödemesini yeniden görüşecek" şekline dönüşür. Sözlüğümüz de kelimeler de aşağıdaki şekilde olsun.

Sözlük :{ekonomi, kriz, hükümet, imf, banka, faiz, borç}

Doküman:{ ekonomik, krizden, dolayı, hükümet, imf, olan, kredi, borçlarını, borcun, ödemesini, yeniden, görüşecek }

Sözlüğümüzde 7 kelime olduğundan dokümanımızın kelime vektörünün boyutu da 7 olacaktır. Buna göre vektörümüz $D_1=(1,1,1,1,0,0,2)$ şeklinde olacaktır. Burada sözlükte

bulunan “borç” kelimesi, 2 defa geçmektedir. Doküman içinde “borçlarını” ve “borcunu” kelimeleri iki ayrı kelime gibi gözükse de kelimenin kökünde “borç” vardır ve anlam olarak aynıdır. Bu tür sert sessiz yumuşaması içeren ve aynı anlamda olup farklı eklerle türeyen kelimeler için Joker (Will Card) yöntemi kullanılmıştır. Örneğin anahtar kelime, “borç” ise jokeri “borc” olarak atanır. Ya da anahtar kelime “beyin” ise jokeri yine aynı anlamda kullanılan “beyni” kelimesi olabilir. Bu yapıdaki kelimeler için yönetim programı içinde anahtar kelimenin yanında joker kelimeler kullanılmıştır. Bu sayede aynı anlam içeren kelimelerin frekansları daha doğru şekilde hesaplanmıştır.

Dokümanların kelime vektörleri bu şekilde oluşturulduktan sonra vektörler, farklı yöntemlerle ağırlıklandırılır. Bu ağırlıklandırma, Terim frekansına göre yapılmıştır. D_1=(1,1,1,1,0,0,2). Terim Frekansına göre yapılan ağırlıklandırma da kelime sayıları dikkate alınır. Bit frekansına göre yapılan ağırlıklandırma da sayıya bakılmaz kelime var veya yok, ona bakılır. Varsa 1, yoksa 0 yazılır. Bit frekansına göre vektörümüz D_1=(1,1,1,1,0,0,1) şeklinde olur.

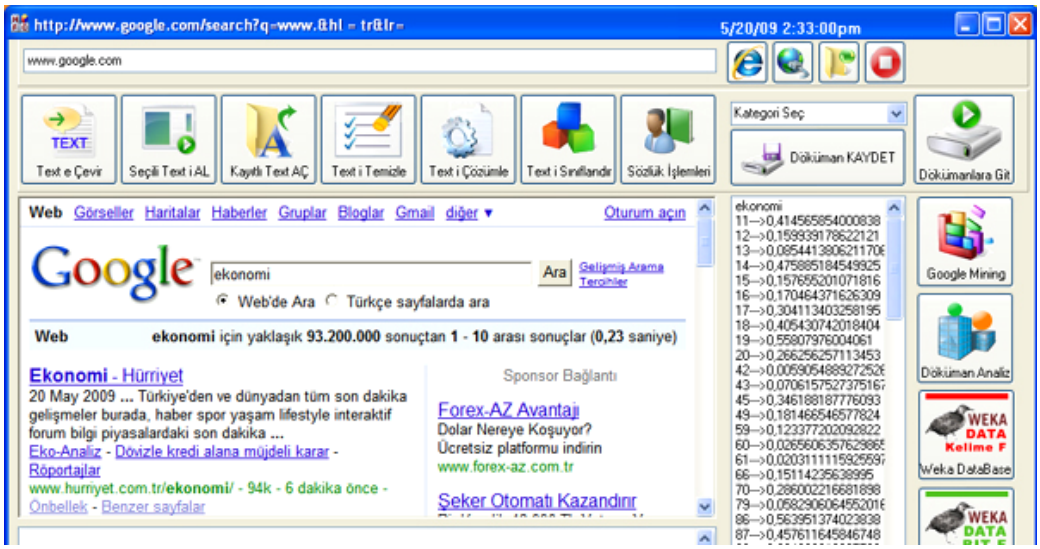
Ağırlıklandırma ne kadar iyi yapılırsa, sınıflandırma da o kadar başarılı olur. Ağırlıklandırma, sınıflandırmanın en önemli aşmalarından biridir. Her sınıflandırma algoritması farklı bir ağırlıklandırma yöntemi kullanabilir. Bu nedenle, farklı vektör ağırlıklandırma yöntemleri geliştirilmiştir.

- Bit Frekansı (BF)
- Terim Frekansı (TF),
- Ters Doküman Frekansı (IDF),
- Terim Frekansı-Ters Doküman Frekansı (TF-IDF),
- Terim Ayrıştırma Değeri, Nitelik Seçme,
- Olasılıksal Terim Ağırlıklandırma,
- Gizli Anlamsal İnceleme (LSI) [5].

Bu çalışma da bit frekansı ve terim frekansına göre vektör ağırlıklandırmaları, yönetim programımız aracılığıyla, diğer vektör ağırlıklandırma ve sınıflandırma işlemleri için WEKA [27] aracı kullanılmıştır.

2.2. Geliştirilen Sistemin Açıklanması

Yapılan çalışma kapsamında metin sınıflandırma, metin çözümleme, doküman analizi ve arama sonuç madenciliği işlemlerini yönetebilmek amacıyla, KTU WDM (KTU Web Data Miner) veri madenciliği yazılımı geliştirilmiştir. KTU WDM, delphi platformunda hazırlanmıştır. Veri tabanı olarak Access veri tabanı ve ADO (Active Data Objects) bağlantıları kullanılmıştır. KTU WDM yazılımı, kendi web tarayıcı bileşeni sayesinde sitelere bağlanıp, istenilen sitede veri madenciliği işlemlerini gerçekleştirebilir. Ayrıca istenilen doküman, seçilen kategori ile birlikte veri tabanına kaydedilip daha sonra eğitim dokümanı olarak da kullanılabilmektedir. KTU WDM de 5 kategori (ekonomi, siyaset, sağlık, spor, bilişim) için yaklaşık 600 eğitim dokümanı kullanılmıştır.



Şekil 9. Ktu Wdm yönetim programı kullanıcı arayüzü

KTU WDM genel olarak aşağıdaki bileşenlerden oluşur.

1. Web Tarayıcı Modülü
2. Metin Çözümleme Araçları
3. Sözlük İşlemleri Modülü
4. Metin Sınıflandırma Araçları
5. Doküman Veri Tabanları Modülü
6. Eğitim Dokümanı Analiz Modülü
7. Arama-Sonuç Madenciliği Modülü (Google Mining)

8. WEKA için, veri tabanı oluşturma bileşeni.

Yukarıda belirtilen 8 bileşen, programımızın ana hatlarını oluşturmaktadır. Bunun yanında ana bileşenler içinde kullanılan diğer araçlara, ilgili kısımlar da ayrıca değinilecektir. Şimdi programın ana bileşenlerini açıklayarak KTU WDM'i daha detaylı inceleyelim.

2.2.1. Web Tarayıcı Modülü

KTU WDM, kendi üzerinde bulunan web tarayıcı bileşeni ile web sayfalarına bağlanır ve sınıflandırma işlemini online olarak gerçekleştirir. Bunun yanında adres satırına yazılan anahtar kelimeyi Google arama motorunda arayarak, arama sonuçlarını sınıflandırır.

Şekil 10. Yönetim programı web tarayıcı bileşeni

Burada ;

 Simgesi, adres satırına yazılan web adresine bağlanır.

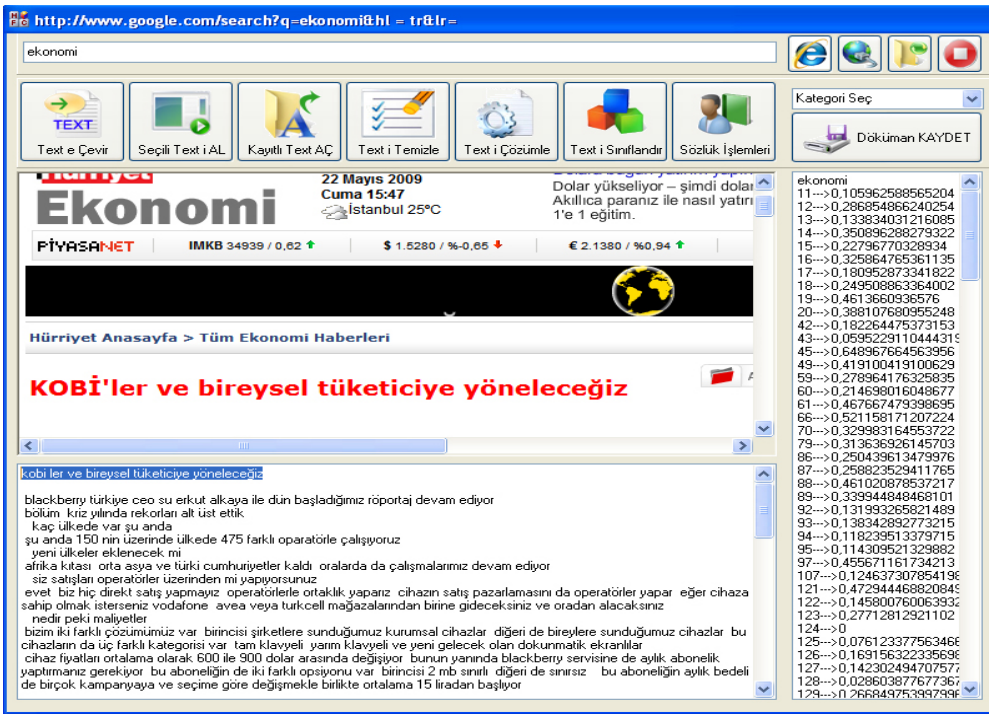
 Simgesi, adres satırına yazılan anahtar kelimeyi, Google arama motorunda arayarak sonuçları listeler.

 Simgesi, bilgisayarda kayıtlı olan web sayfalarını yükleyerek, kayıtlı dosyalar üzerinde sınıflandırma yapmamızı sağlar.

 Simgesi, web sayfasının yüklenmesini durdurur.

2.2.2. Metin Çözümleme Araçları

Web tarayıcı ile bağlandığımız web sayfaları, çok farklı bileşenlerden oluşabilmektedir. Web sayfaları, resim, metin, animasyon, video, flash ve benzeri bileşenler içerir. Web tarayıcı da görüntülenen sayfanın metinleri, "Text'e Çevir" düğmesi tıklanarak tarayıcının altında bulunan metin kutusuna, saf metin olarak aktarılır. Bu aktarma işlemi esnasında metin tamamen küçük harfe dönüştürülür ve noktalama işaretlerinden arındırılır.



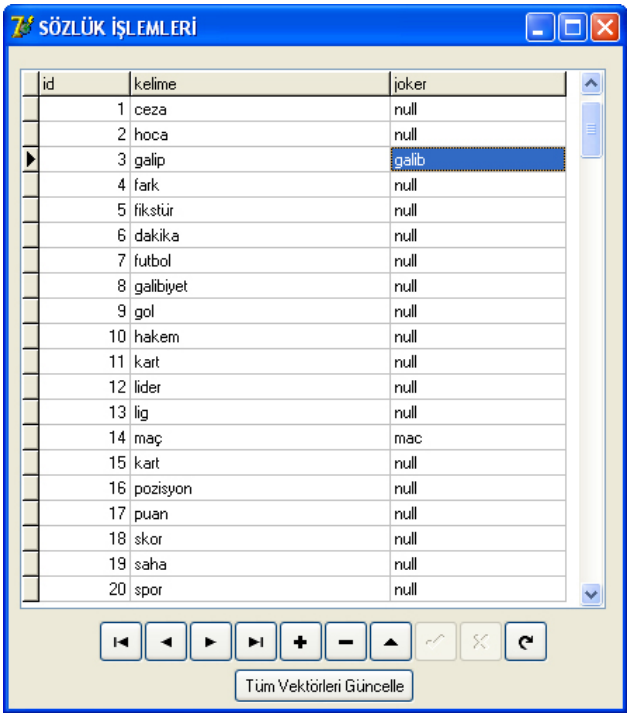
Şekil 11. Metinlerin çözümlenmesi ve sınıf tespiti

Çözümleme işlemi aktarma sırasında tamamlanır. Eğer dışarıdan metin kutusuna kopyalanan bir metin olursa, bu durumda "Text'i Çözümle" düğmesine tıklanarak metin çözümlenir. Metin kutusunda ki saf metin, bu aşamadan sonra "Text'i Sınıflandır" düğmesine tıklanarak, tanımlanmış sınıflara olan yakınlığı tespit edilir. Bu tespit, şu şekilde yapılır. Metin kutusuna aktarılmış ve çözümlenmiş metin veri tabanında kayıtlı eğitim dokümanları ile tek tek karşılaştırılır. Bu karşılaştırma K-NN algoritmasına göre yapılır. Sonuçlar şekilde görüldüğü gibi listelenir. Listede test edilen metnin, tüm eğitim dokümanları ile benzerliği sayısal olarak gösterilir. Ayrıca her kategorinin maksimum

değeri belirlenir ve listede özet bilgi olarak yer alır. Bu şekilde metnin ait olduğu kategori belirlenmiş olur. Bu aşamadan sonra, kategorisi belirlenen metin, eğer istenirse "Doküman KAYDET" düğmesi tıklanarak veri tabanına kaydedilir. Kaydedilen doküman sonradan test veya eğitim dokümanı olarak kullanılır.

2.2.3. Sözlük İşlemleri Modülü

Sözlük, metinlerde aradığımız anahtar kelimelerle ilgili işlemleri içeren bileşendir. Şekilde de görüldüğü gibi her kelimenin bir id numarası vardır. Bunun yanında aynı anlam içeren kelimeler için joker ler kullanılmıştır. Programımızın sözlük bölümü, 5 kategori (ekonomi, siyaset, sağlık, spor, bilişim) için 300 anahtar kelime içermektedir. Bu da vektör boyutumuzun 300 olacağı anlamına gelmektedir.



Şekil 12. Sözlük işlemleri kullanıcı ara yüzü

Buna ek olarak kullanıcı istediği zaman kelimeler üzerinde değişiklikler yapabilir. Kelime ekleme, kelimeye joker atama, kelime silme, kelime değiştirme gibi işlemler yine sözlük bölümünden yapılır. Eğer sözlükte değişiklikler yapılırsa kayıtlı metinlere ait kelime vektörlerinin, sözlükteki değişikliklere göre güncellenmesi gerekecektir. Bunun için

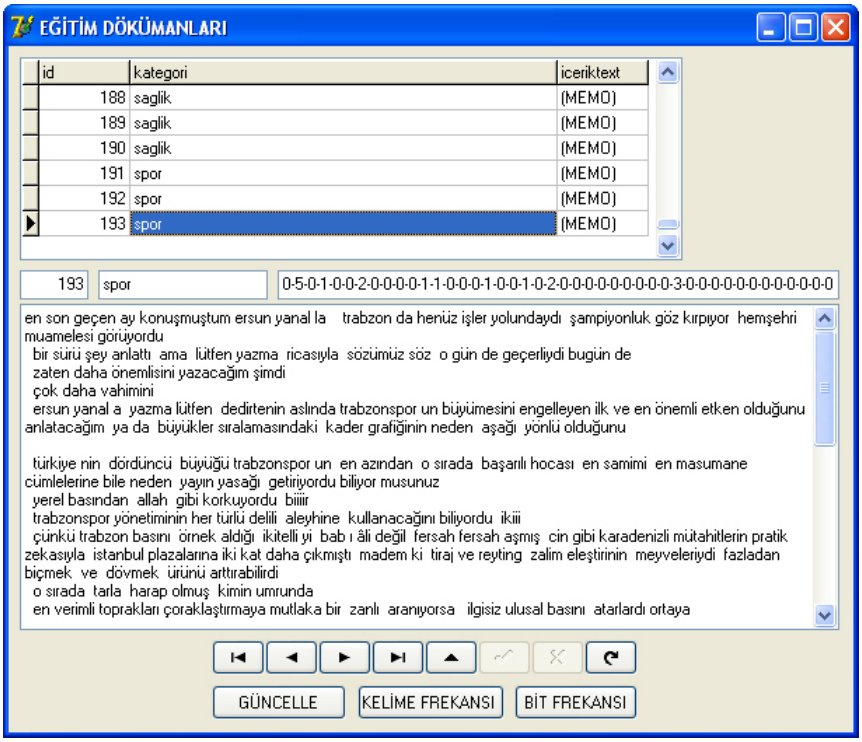
sözlük işlemlerinden sonra şekilde görülen “Tüm Vektörleri Güncelle” düğmesi tıklanarak, vektörlerin güncellenmesi sağlanır.

2.2.4. Metin Sınıflandırma Araçları

Metin sınıflandırma işlemi, online olarak web üzerinde veya veri tabanına kayıtlı dokümanlar üzerinde yapılabilmektedir. Metin çözümleme kısmında da bahsedildiği gibi metinlerin sınıflandırma işlemi şu şekilde yapılır. Program da, metin kutusuna aktarılmış ve çözümlenmiş metin veri tabanında kayıtlı eğitim dokümanları ile tek tek karşılaştırılır. Bu karşılaştırma K-NN algoritmasına göre yapılır. Sonuçlar liste kutusunda her eğitim dokümanı için listelenir. Listede, test edilen metnin, tüm eğitim dokümanları ile benzerliği sayısal olarak gösterilir. Ayrıca her kategorinin maksimum değeri belirlenir ve listede özet bilgi olarak yer alır. Bu şekilde metnin ait olduğu kategori belirlenmiş olur. Bu aşamadan sonra, kategorisi belirlenen metin, eğer istenirse “Doküman KAYDET” düğmesi tıklanarak veri tabanına kaydedilir. Kaydedilen doküman sonradan test veya eğitim dokümanı olarak kullanılır. KTU WDM’ metin sınıflandırma işlemleri için K-NN algoritmasını kullanır. Karar ağaçları, SVM algoritması, Navie Bayes olasılık metodu, gibi diğer sınıflandırma algoritmaları ve bunların farklı modellerinin uygulanması için WEKA[27] aracı kullanılmıştır. Bu aşama da WEKA için gerekli, kelime vektörlerinin bulunduğu veri tabanı dosyaları KTU WDM tarafında oluşturulur.

2.2.5. Doküman Veri Tabanı Modülü

Sistemi eğitmekte kullandığımız tüm dokümanlara buradan erişim sağlanır. Programın ana penceresinden “Dokümanlara Git” düğmesi tıklanarak bu bölüme ulaşılır. Bu bölümde eğitim dokümanlarının kayıt numarası, kategori bilgisi, içerik metni ve kelime vektörü bilgileri tutulur. Metinlerin içeriğini ve kelimelerin frekansını gösteren, kelime vektörünü buradan görebiliriz. Metin üzerinde herhangi bir değişiklik yapılırsa, “Güncelle” düğmesi tıklanarak, metnin son haline göre kelime vektörleri güncellenir.



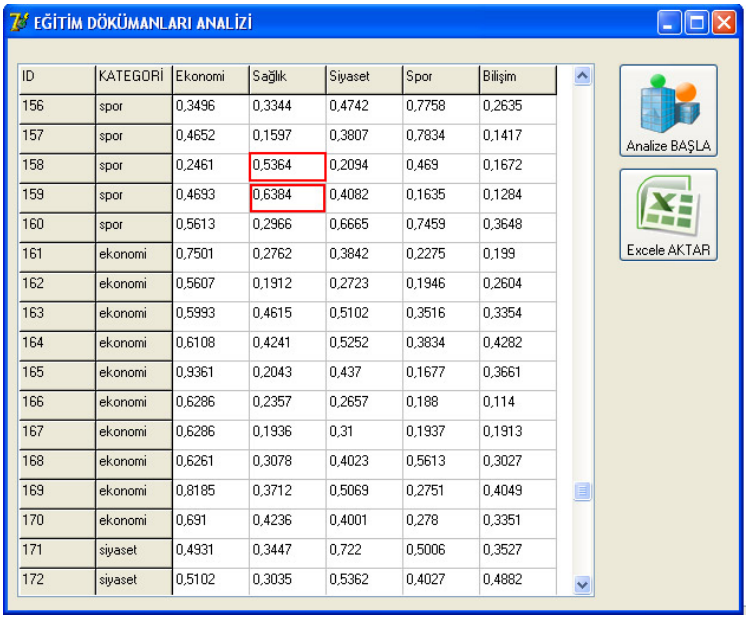
Şekil 13. Eğitim dokümanları ve vektör tabloları veritabanı ara yüzü

Ayrıca dokümanların kelime vektörlerini, kelime frekansı veya bit frekansına göre ağırlıklandırarak WEKA için veri dosyası bu kısımdan oluşturulur. Oluşturulan dosyalar WEKA için kullanılan ham veri dosyalarıdır. Bu sayede buradan oluşturulan dosyalar hiçbir ön işlem gerektirmeden WEKA' ya uygulanıp, rahatlıkla sınıflandırma algoritmaları test edilir.

2.2.6. Doküman Analiz Modülü

Doküman analizi, eğitim dokümanlarının tüm kategorilere yakınlığını analiz eder. Şekilde de görüldüğü gibi her dokümanın kayıt numarası ve atandığı kategori listelenir. "Analiz'e Başla" düğmesi tıklandığında her dokümanın kategorilere olan yakınlığı sayısal olarak hesaplanır ve tabloda listelenir. Örneğin 158. ve 159. dokümanlar, spor dokümanı olmasına karşın sağlık değeri daha yüksektir. Demek ki metinler, spor dokümanı olmasına karşın, içerik olarak sağlık sınıfına yakın olduğu anlaşılıyor. Bu şekilde metinlere atadığımız kategorilerin doğruluğunu, sayısal olarak değerlendirmiş oluyoruz. Şekilde kırmızı çerçeve içine alınmış sonuçlar da, atanan kategori ile sayısal değerlerin işaret ettiği kategoriler farklılık göstermektedir. Bu sonuçlar da algoritmaların başarı yüzdesi içinde

hata payı olarak gözükecektir. İleri de sonuç ve tablolar kısmında hatalı metin örnekleri daha detaylı verilecektir.



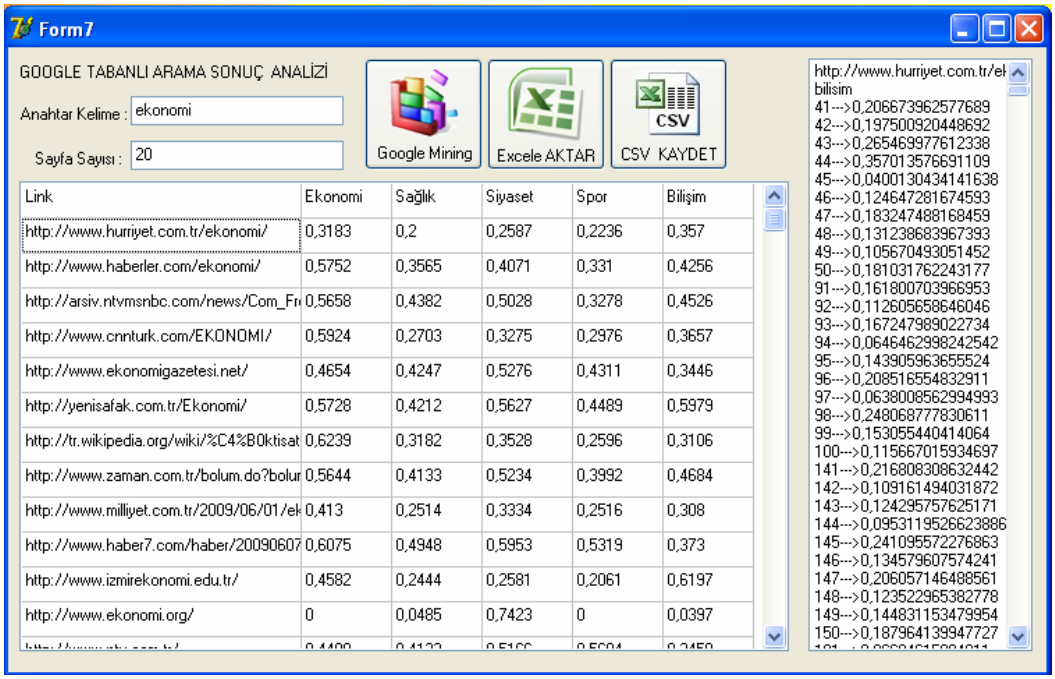
ID	KATEGORİ	Ekonomi	Sağlık	Siyaset	Spor	Bilişim
156	spor	0,3496	0,3344	0,4742	0,7758	0,2635
157	spor	0,4652	0,1597	0,3807	0,7834	0,1417
158	spor	0,2461	0,5364	0,2094	0,469	0,1672
159	spor	0,4693	0,6384	0,4082	0,1635	0,1284
160	spor	0,5613	0,2966	0,6665	0,7459	0,3648
161	ekonomi	0,7501	0,2762	0,3842	0,2275	0,199
162	ekonomi	0,5607	0,1912	0,2723	0,1946	0,2604
163	ekonomi	0,5993	0,4615	0,5102	0,3516	0,3354
164	ekonomi	0,6108	0,4241	0,5252	0,3834	0,4282
165	ekonomi	0,9361	0,2043	0,437	0,1677	0,3661
166	ekonomi	0,6286	0,2357	0,2657	0,188	0,114
167	ekonomi	0,6286	0,1936	0,31	0,1937	0,1913
168	ekonomi	0,6261	0,3078	0,4023	0,5613	0,3027
169	ekonomi	0,8185	0,3712	0,5069	0,2751	0,4049
170	ekonomi	0,691	0,4236	0,4001	0,278	0,3351
171	siyaset	0,4931	0,3447	0,722	0,5006	0,3527
172	siyaset	0,5102	0,3035	0,5362	0,4027	0,4882

Şekil 14. Eğitim dokümanları analiz ve sınıf doğrulaması

Ayrıca sonuçlar, “Excele Aktar” düğmesi tıklanarak excel dosyası olarak saklanır ve daha sonra excel de, sayısal ve grafiksel olarak analiz edilebilir.

2.2.7. Arama Sonuç Madenciliği Modülü

Veri madenciliğinin en fazla gereksinim duyulan kısmı sanırım burasıdır. Çünkü verilerin sınıflandırılması ile ilgili hali hazırda ciddi bir çalışma yoktur. Her kullanıcı bir arama motorunda arama yapar. Fakat aramadan elde edilen web sayfalarının çoğu gerçekte bizim aradığımız konuyla ilgisi bile yoktur. Bu da zaman kaybına ve doğru veriden uzaklaşmaya yol açar. Arama motoru ile birlikte çalışacak bir filtreleme bileşeni ile bu sorun kısmen de olsa aşılabilir. Bizim burada yapmak istediğimiz, programda belirlenmiş olan ekonomi, siyaset, sağlık, bilişim ve spor kategorileri için, google arama motoru ile bütünleşik bir konu sınıflandırma geliştirmektedir. Sistemi denemek amacıyla şimdilik 5 kategori kullanılacaktır. İstenildiği zaman kategoriler ve anahtar kelimeler değiştirilerek program daha da genişletilebilir. Bunun için yapılacak şey sisteme kategori, kelime ve doküman eklemektir.



Şekil 15. Google tabanlı arama sonuç analizi

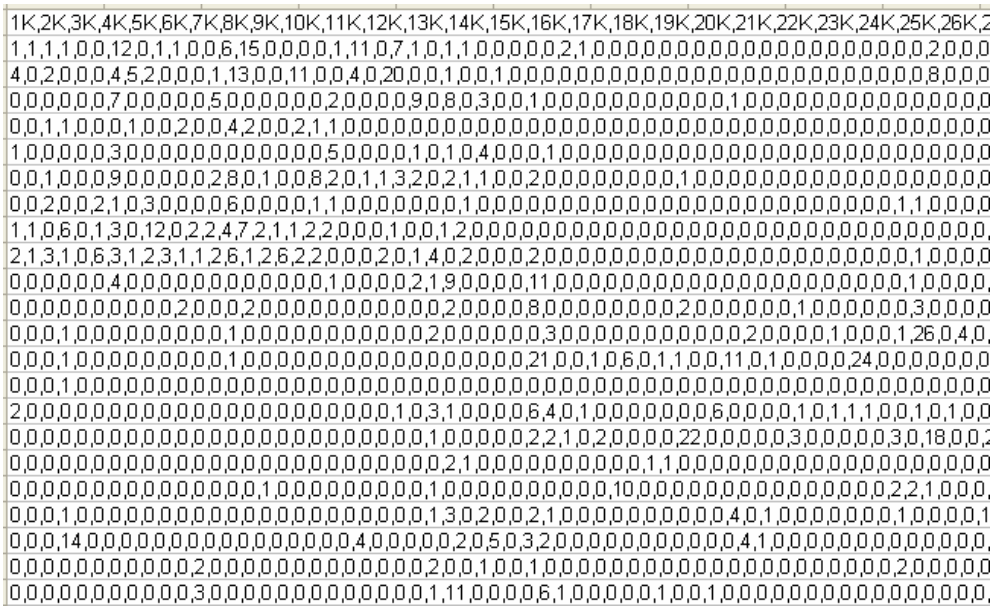
Bu bölümde, Google arama motoru ile bütünleşik olarak geliştirilen konu sınıflandırma sistemi, Şekil 15. te detaylarıyla gösterilmiştir. Sistemi açıklayacak olursak, ilk önce, google da aranacak anahtar kelime girilir. Sonuçlardan kaç tanesinin analiz edileceği sayfa sayısı kısmına girilir ve analiz başlatılır. Google da anahtar kelime aranır. Elde edilen sonuçların sayfa sayısı ile belirlenen kısmı analiz edilir. Örneğin sonuçlardan ilk 20 tanesi. Google da bulunan her sayfa tek tek analiz edilir. Bulunan sonuçlar, yani sayfaların kategorilere yakınlığı, sayfanın adresi ile birlikte kullanıcıya listelenir. Şekilde ki örnekte ekonomi ile ilgili aranan sayfaların ilk 20 tanesinin ekonomi ye ve diğer kategorilere yakınlığı analiz edilmiştir.

2.2.8. Weka İçin, Veri Tabanı Oluşturma Modülü

Weka [27], veri madenciliği alanında yaygın olarak kullanılan bir yazılımdır. Fakat kullanım ile ilgili bazı sıkıntılar mevcuttur. Weka, programı her türlü dosyayı okuyamamaktadır. Weka da, verilerin analiz edilebilmesi için verilerin, Weka' ya uygun formatta düzenlemesi gerekir. Weka nın desteklediği dosya türleri arff, csv, xrff, sql dosya türleridir. Bu yüzden verilerin bu dosya türlerinden herhangi birisine uygun olması gerekir. Yoksa, farklı formattaki dosyaları Weka da çalıştırmak olanaksızdır. Bu sorunu aşmak

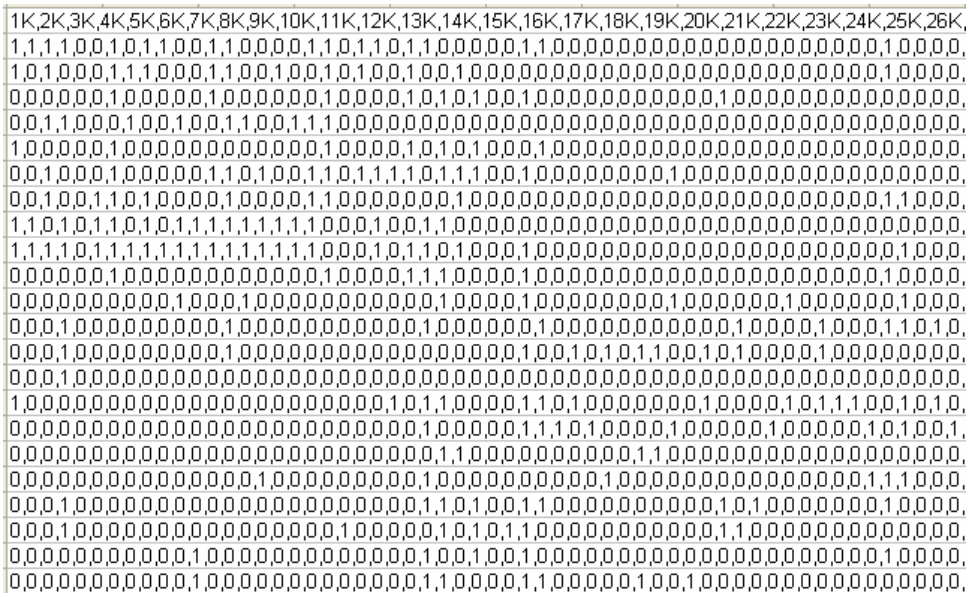
için, Weka veri dosyaları programımız tarafından oluşturulur. Oluşturulan dosyalar, başka hiçbir işleme gerek duymadan direk olarak Weka da çalıştırılabilir. Weka için iki tür veri dosyası oluşturulur.

Weka için oluşturulan dosya türlerinden ilki, yandaki düğme tıklanarak oluşturulan kelime frekansına bağlı vektör tablosudur. Bu dosya türünde kelime sayıları dikkate alınır. Her kelimenin sayısal olarak frekansı verilir. Vektör tabloları ve veri dosyası buna göre oluşturulur. Oluşturulan dosya csv formatındadır.



Şekil 16. Kelime frekansına göre oluşturulmuş vektör tablosu

Weka için oluşturulan diğer dosya türü ise, yandaki düğme tıklanarak oluşturulan bit frekansına bağlı vektör tablosudur. Bu dosya türünde kelime sayıları dikkate alınmaz. Kelimenin var olup olmadığına bakılır. Varsa 1, yoksa 0 kodlanır. Vektör tabloları ve veri dosyası buna göre oluşturulur. Oluşturulan dosya csv formatındadır.



Şekil 17. Bit frekansına göre oluşturulmuş vektör tablosu

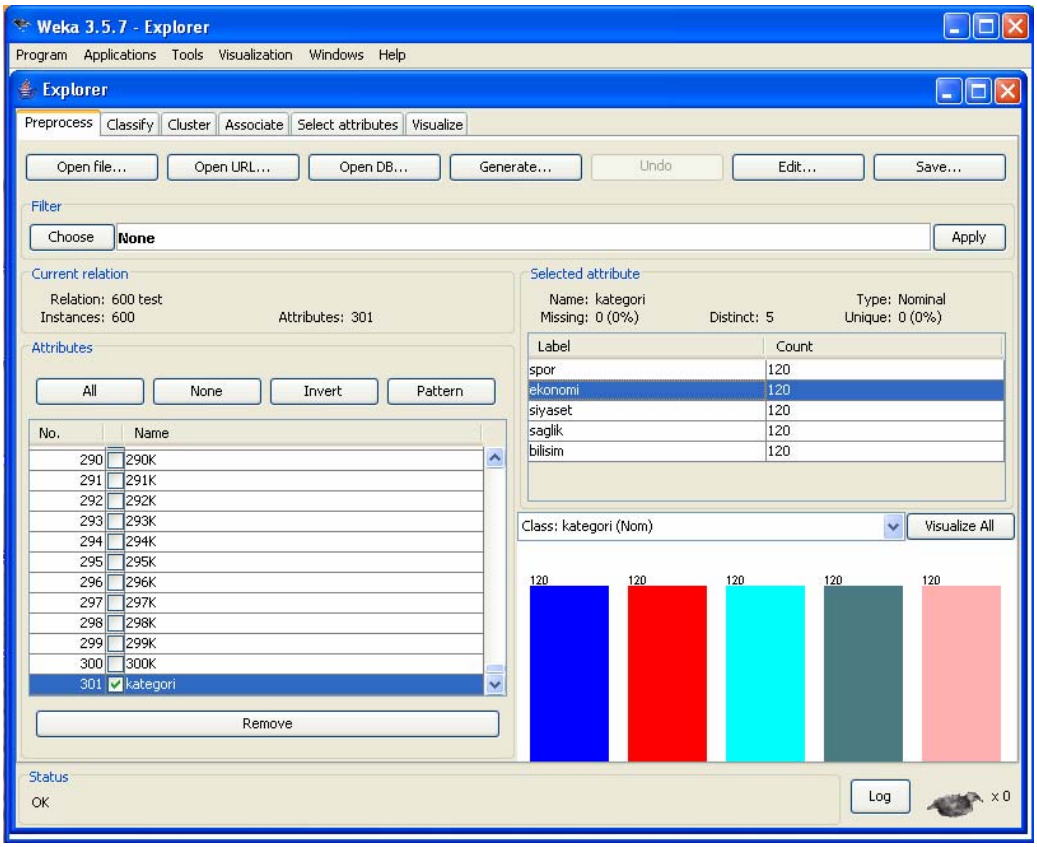
2.3. Weka Aracının Kullanımı

WEKA, Waikato Üniversitesinde bir proje olarak başlayıp, bugün dünya üzerinde birçok insan tarafından kullanılmaya başlanan bir Veri Madenciliği uygulaması geliştirme programıdır. Temel veri madenciliği işlemlerinin rahatlıkla uygulanabildiği bir araç olan WEKA, görsel ara yüzüyle büyük bir kullanım kolaylığına sahiptir. Java platformu üzerinde geliştirilmiş açık kaynak kodlu bir programdır. Sınıflandırma, kümeleme ve veri ilişkilendirme alanlarında zengin bir içeriğe sahiptir**.** Veri sınıflandırma algoritmaları, veri kümeleme algoritmaları, veri ilişkilendirme algoritmaları ve bu algoritmaların sayısal ve grafiksel olarak gösterimleri mevcuttur.

Açık kaynak kodlu olduğu için sürekli güncellenen bir içeriğe sahiptir. Bu yüzden WEKA programı son 1-2 yıl içerisinde büyük bir gelişim göstermiştir. Çalışma kapsamında Weka 3.5.7 sürümü kullanılmış olup, uygulanan algoritmalar ve alınan sonuçlar performans kriterlerine göre yorumlanmıştır. Ana giriş penceresi aşağıdaki şekilde verilen programın, seçilen veri dosyası ile bilgiler ana ekranda gösterilmektedir. Veri dosyası, Preprocess → Open file komutlarıyla açıldıktan sonra, yapılacak işlemler için ilgili sekmelere tıklanır. Şekildeki örnekte 600 dokümanlık bir veri kümesini açıklayıcı bilgiler yer almaktadır. Şekle göre 5 kategori (ekonomi, siyaset, sağlık, spor, bilişim) olduğu ve her kategoriye ait 120 doküman olduğu görülmektedir. Attributes : 301 olarak

gözüken kısım, ise belirleyici özellik sayısını yani vektörlerin boyutunu vermektedir. Diğer bir deyişle sözlükteki tanımlayıcı kelime sayısını göstermektedir.

Sözlükte 300 kelime vardı, buna birde her vektörün kategorisi eklenirse 301 tane belirleyici özellik tanımlanmış olur.



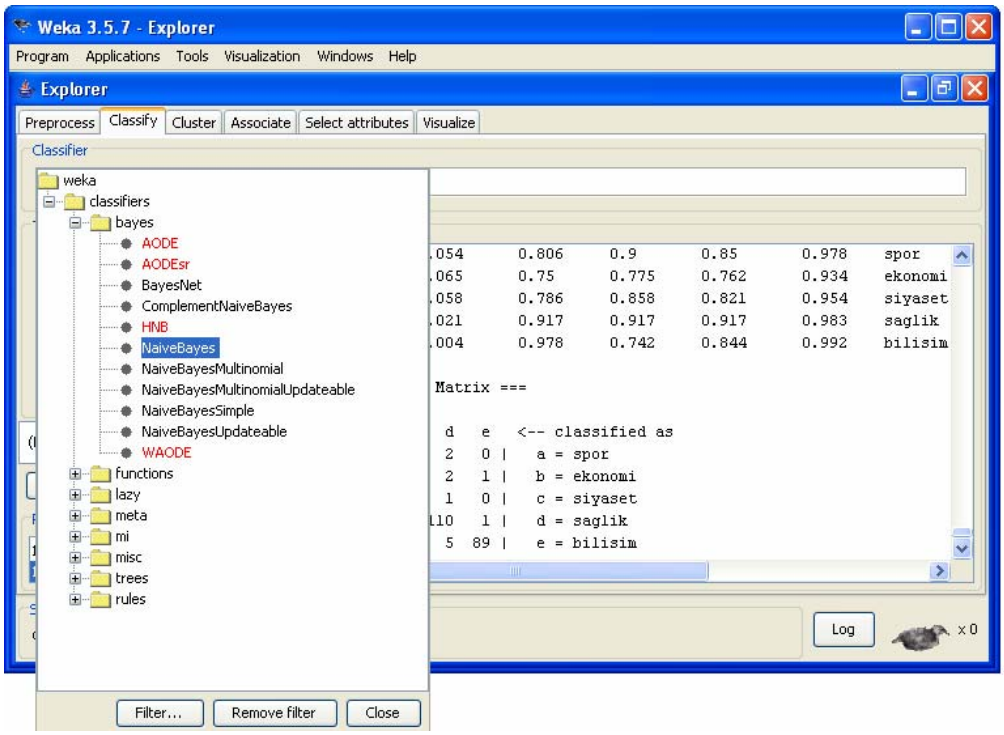
Şekil 18. Weka yazılımı kullanıcı ara yüzü

Veri dosyası, Preprocess → Open file komutlarıyla açıldıktan sonra, yapılacak işlemler için ilgili sekmelere tıklanır. Sınıflandırma için Classify, Kümeleme için Cluster, İlişkilendirme için Associate seçilir. Özellik Seçimi için "Select Attributes", algoritma sonuçlarının grafiksel gösterimi için "Visualize" sekmeleri seçilir.

2.3.1. Weka ile Veri Sınıflandırma

Daha önce KTU WDM programı ile sayısallaştırıp, kelime vektörleri haline getirilen metin dokümanları csv formatında veri dosyası olarak kaydedilir. Bu aşamadan sonra, rahatlıkla sınıflandırma işlemi uygulanabilir. Bir önceki adımda açıkladığımız 600 dokümanı içeren veri dosyasını sınıflandıralım. Veri dosyasının açılmasını ve

yüklenmesini bir önceki konuda bahsetmiştik. Veri dosyası yüklendikten sonra sınıflandırma işlemi için "Classify" sekmesi tıklanır. Aşağıdaki ekran görüntüsü şeklinde de görüldüğü gibi, listeden bir sınıflandırma algoritması seçilir. Örnek te Naive Bayes sınıflandırma algoritması seçilmiştir.

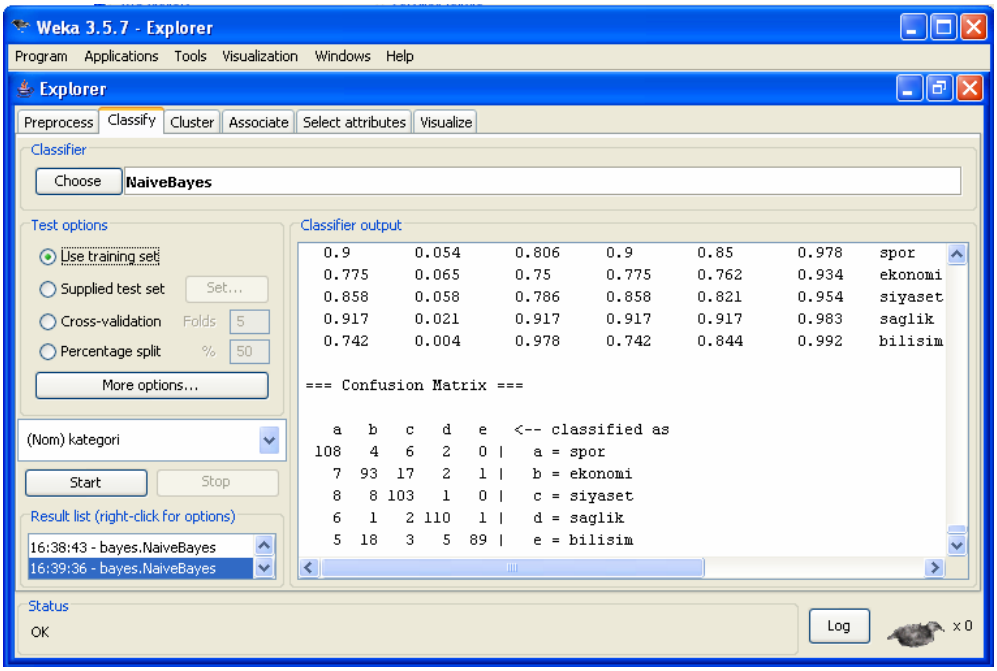


Şekil 19. Weka yazılımı veri sınıflandırma ara yüzü

Sınıflandırma algoritması "Naive Bayes" olarak seçildikten sonra, algoritma ile ilgili test seçenekleri belirlenir. Bu seçenekler, Sınıflandırıcı algoritmayı seçtiğimiz kısmın hemen altında "Test options" kısmında verilmektedir. Bu seçenekleri açıklayacak olursak;

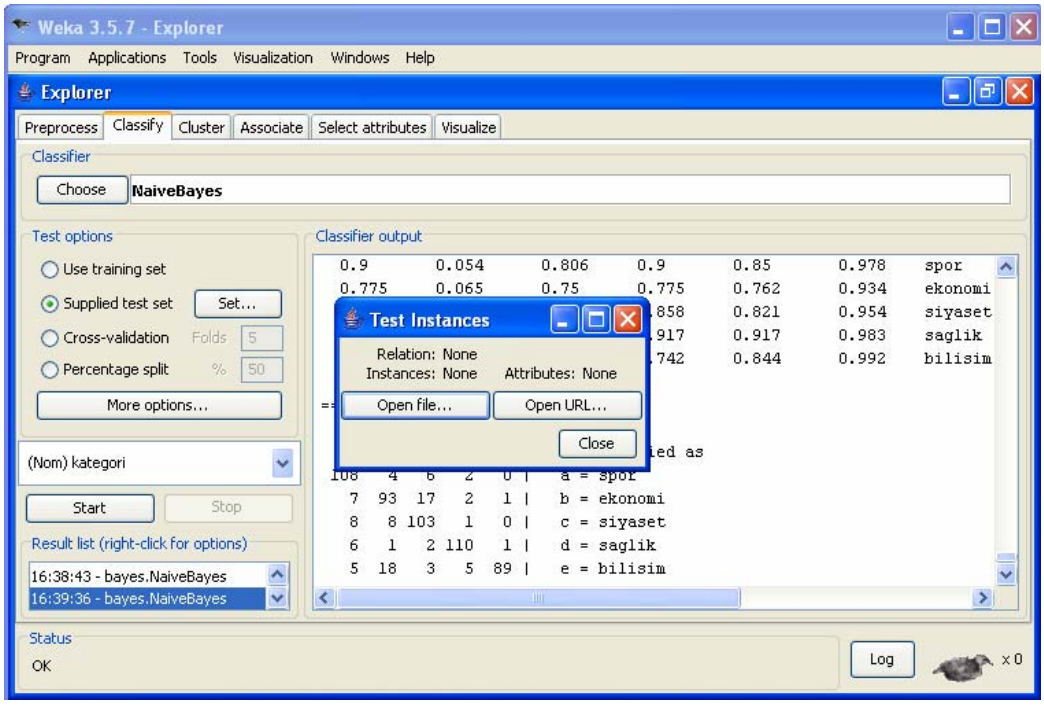
Eğitim Seti Kullan (Use training set) : Bu seçenekte tüm dokümanlar eğitim dokümanı olarak kullanılır ve kendi içinde birbirleriyle karşılaştırılarak sınıflandırılır. Burada eğitim kümesi de test kümesi de aynıdır. Bununla ilgili detaylar aşağıdaki şekilde verilmiştir. Şekle bakılacak olursa 600 dokümandan oluşan verimiz bu seçenek ile, kendi içinde sınıflandırılmış ve kategorilere göre yanlış olan sınıfların hata matrisi şekilde verilmiştir. Her kategori için 120 doküman vardı. Hata matrisine bakılırsa en başarılı kategori, sağlık (110/120) kategorisi olarak gözükmektedir. Tüm kategoriler için bakılırsa 600 dokümandan 503 tanesinin atandığı sınıf doğru, 97 dokümanın sınıfı yanlış gözükmektedir. Naive Bayes algoritmasına göre bu 600 dokümanlık eğitim kümesinin

sınıflandırma başarısı % 83,8 olarak hesaplanır. Demek ki yaptığımız sınıflandırma, Naive Bayes algoritmasına göre %83,8 oranında başarılıdır.



Şekil 20. Weka da sınıflandırma algoritmalarının uygulanması

Test Setini Kullan (Supplied test set) : Bu seçenek ile Preprocess → Open file komutu ile açılan dosya eğitim seti olarak kullanılır. Sistem bu veri kümesi ile eğitilir. Set düğmesi tıklanarak gelen pencereden “Open file” düğmesi tıklanır. Buradan test verisi açılır ve daha önceden eğitilmiş sistemde veriler test edilir. Bu işlemler le ilgili adımlar, aşağıdaki şekilde verilen ekran görüntüsün de detaylı olarak gösterilmiştir. Burada hem seti ve test seti ayrı iki kümeyi temsil eder. Sistem önce eğitilir, sonra test verileriyle test edilir. Bu kullanım ile ilgili daha detaylı sonuç ve istatistikler, ileri bölümlerde sonuçlar kısmında detaylı olarak verilecektir.



Şekil 21. Eğitim seti ve test setinin modellenmesi

Çapraz Doğrulama (Cross Validation): Bu kullanım bir önceki adımda anlattığımız eğitim seti ile test setinin birlikte kullanılmasına benzer. Çapraz doğrulama tüm veriyi kullanmayı sağlayan bir metottur. Veri seti rasgele 2 eşit kısma ayrılır. İlk veri setiyle sistem eğitilir, eğitilen sistem diğer veri setiyle test edilir ve doğruluk oranı hesaplanır. Daha sonra test veri setiyle sistem eğitilir ve eğitilen sistem birinci veri setiyle tahmin edilip, doğruluk oranı hesaplanır. Ve en sonunda tüm veriler kullanılarak model oluşturulur. Daha önceden hesaplanmış olan 2 doğruluk oranının ortalamasıyla, son model karşılaştırılır. n kere çapraz doğrulama (n Fold Cross Validation) da ise veri seti rasgele n gruba ayrılır. 1. grup test için ayrılırken geriye kalan n-1 grupla sistem eğitilir. Eğitilen sistem, test için ayrılan veriler üzerinde test edilir ve doğruluk oranı hesaplanır. Süreç n defa tekrar eder ve modelin doğruluk oranı, n tane doğruluk oranının ortalaması kadar olur. Weka da, çapraz doğrulamaya göre sınıflandırma yapılmak istenirse, yukarıdaki şekilde görüldüğü gibi "Cross validation" seçilir ve "Folds" kısmına bir sayı girilir. Veri seti kaç alt kümeye bölünmek isteniyorsa sayı ona göre belirlenir ve sınıflandırma işlemi gerçekleştirilir.

Yüzdelik Dilim (Percentage Split) : Bu seçenek eğitim seti ile test seti ni yüzde olarak böler ve sınıflandırır. Örneğin değer olarak %70 girdiğimizi varsayarsak tüm veri setinin %70 ini eğitim seti, kalan %30 unu ise, test seti olarak kullanır. Bu yüzdelere göre model oluşturulur ve sınıflandırma bu yüzdelere göre yapılır.

3. SONUÇLAR

3.1. KNN Algoritmasında K' Komşuluk Değerinin Performansa Etkisi

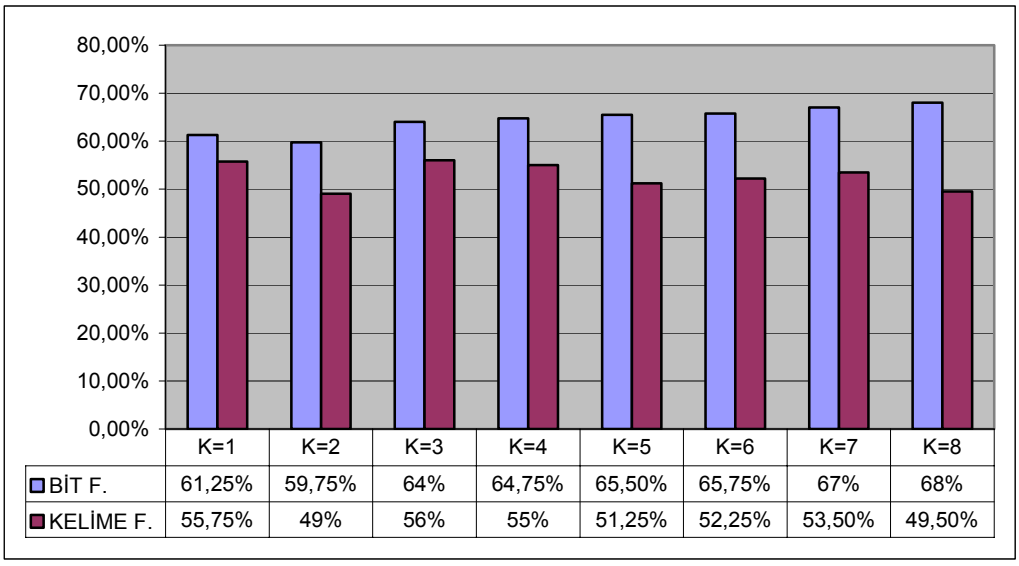
Farklı K komşuluk değerleri denenerek 200 eğitim ve 400 test dokümanı ile KNN Algoritması için sınıflandırma ölçümleri yapılmış ve K- en yakın komşuluk değerinin performansa etkisi incelenmiştir. Tablo 1. de bit frekansına göre en başarılı değer K=8, Tablo 2. de ise kelime frekansına göre en başarılı değer K=3 komşuluğu için gerçekleşmiştir.

Tablo 1. KNN algoritmasının, K-komşuluk değerleri için performansı (bit f.)

K	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
BAŞARI	%61,25	%59,75	%64	%64,75	%65,5	%65,75	%67	%68

Tablo 2. KNN algoritmasının, K-komşuluk değerleri için performansı (kelime f.)

K	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8
BAŞARI	%55,75	%49	%56	%55	%51,25	%52,25	%53,5	%49,5



Şekil 22. KNN algoritması performans grafiği

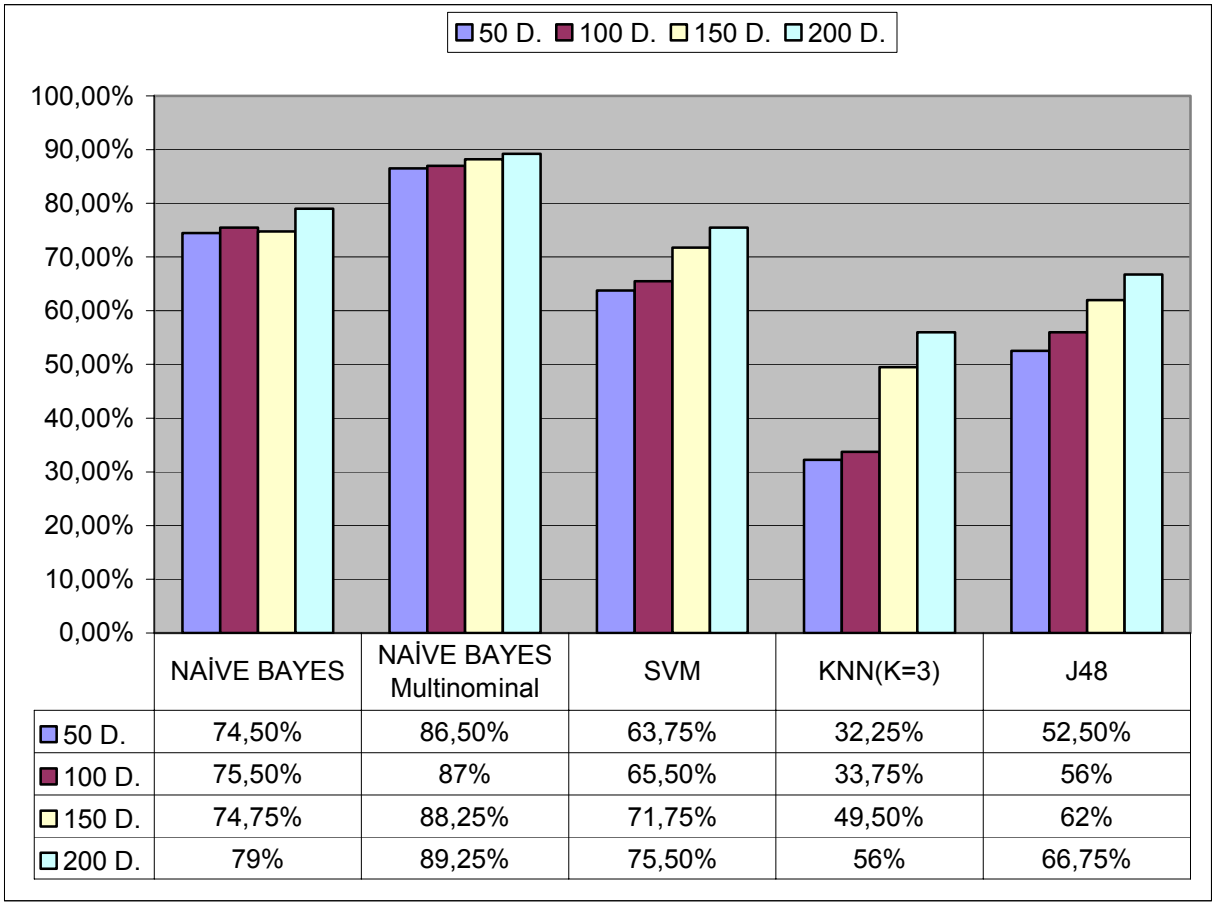
3.2. Eğitim Dokümanı Sayısının Performansa Etkisi

Eğitim dokümanı sayısının, veri sınıflandırma başarısını etkileyen bir faktör olduğunu daha önceki bölümde belirtmiştik. Doküman sayısının farklı algoritmalar üzerinde performansa olan ektisini değerlendirmek için, farklı sayıda eğitim setleri kullanılarak dokuman sayısının performansa etkisi incelenmiştir. Eğitim seti, 50, 100, 150 ve 200 dokümanlık setlerle test edilmiş, kelime frekansı ve bit frekansı ağırlığına göre, 400 dokümanlık test seti için, farklı sınıflandırma algoritmaları üzerinde doküman sayısının, performansa etkisi incelenmiştir. Tablo 3. de kelime frekansına göre algoritmaların doküman sayısına bağlı performans değerleri verilmiştir.

Tablo 3. Doküman sayısına göre kelime frekansı için algoritmaların başarı değerleri

Algoritmalar	50 D.	100 D.	150 D.	200 D.	ORT.
NAİVE BAYES	%74,5	%75,5	%74,75	%79	%75,94
NAİVE BAYES Multinominal	%86,5	%87	%88,25	%89,25	%87,75
SVM	%63,75	%65,5	%71,75	%75,5	%69,13
KNN(K=3)	%32,25	%33,75	%49,5	%56	%42,88
J48 (Karar Ağacı)	%52,5	%56	%62	%66,75	%59,31

Ayrıca her sınıflandırma algoritması için ortalama performans değerleri de ölçülmüş olup yine Tablo 3. de verilmiştir. Elde edilen sonuçlara dayanarak şu yorumu yapabiliriz. Eğitim dokümanı sayısı arttıkça algoritmaların sınıflandırma başarısı da artmıştır. Sistem ne kadar fazla sayıda dokümanla eğitilirse elde edilen sınıflandırma başarısı da o oranda artacaktır.



	NAİVE BAYES	NAİVE BAYES Multinominal	SVM	KNN(K=3)	J48
50 D.	74,50%	86,50%	63,75%	32,25%	52,50%
100 D.	75,50%	87%	65,50%	33,75%	56%
150 D.	74,75%	88,25%	71,75%	49,50%	62%
200 D.	79%	89,25%	75,50%	56%	66,75%

Şekil 23. Doküman sayısına göre kelime frekansı için algoritmaların başarı grafiği

Eğitim dokümanı sayısına göre Şekil 23. de gösterilen grafiği açıklayacak olursak, eğitim dokümanı sayısı arttıkça tüm algoritmaların başarısı gözle görülür oranda artmıştır. Bunun yanında algoritmaların başarı yüzdesi olarak bakıldığında, sınıflandırma başarısı en yüksek algoritma, Naive Bayes Multinominal modeldir. Çok yüksek oranda diğer algoritmalarla performans farkı vardır. Ortalama başarısı %90' a yaklaşırken diğer algoritmalar daha düşük seviyede kalmıştır. Bu sonuçlar, Şekil 22. de değerler tablosu ile birlikte verilmiştir. Ayrıca 200 eğitim, 400 test dokümanı için kelime frekansına göre, algoritmaların başarı değerleri ve sınıflara göre hata matrisleri tablosu Tablo 4. te verilmiştir.

Tablo 4. Kelime frekansı için sınıflara göre hata matrisleri

ALGORİTMA	BAŞARI	HATA MATRİSİ
NAİVE BAYES	**%79 (316/400)**	a b c d e <-- classified as **76** 1 2 1 0 | a = spor 6 54 10 0 10 | b = ekonomi 10 8 59 1 2 | c = siyaset 1 4 1 74 0 | d = saglik 4 16 3 4 53 | e = bilisim
NAİVE BAYES Multinominal	**%89,25 (357/400)**	a b c d e <-- classified as 72 1 2 5 0 | a = spor 2 71 4 1 2 | b = ekonomi 0 7 73 0 0 | c = siyaset 1 2 1 **76** 0 | d = saglik 0 10 2 3 65 | e = bilisim
SVM	**%75,5 (302/400)**	a b c d e <-- classified as 68 1 0 11 0 | a = spor 4 69 2 3 2 | b = ekonomi 8 14 51 7 0 | c = siyaset 3 2 1 **74** 0 | d = saglik 2 9 2 27 40 | e = bilisim
KNN(K=3)	**%56 (224/400)**	a b c d e <-- classified as 51 23 2 4 0 | a = spor 5 **62** 12 1 0 | b = ekonomi 8 27 45 0 0 | c = siyaset 4 16 1 59 0 | d = saglik 1 54 14 4 7 | e = bilisim
J48 (Karar Ağacı)	**%66,75 (267/400)**	a b c d e <-- classified as **58** 19 0 2 1 | a = spor 2 56 15 0 7 | b = ekonomi 1 21 54 4 0 | c = siyaset 0 17 3 **58** 2 | d = saglik 0 31 8 0 41 | e = bilisim

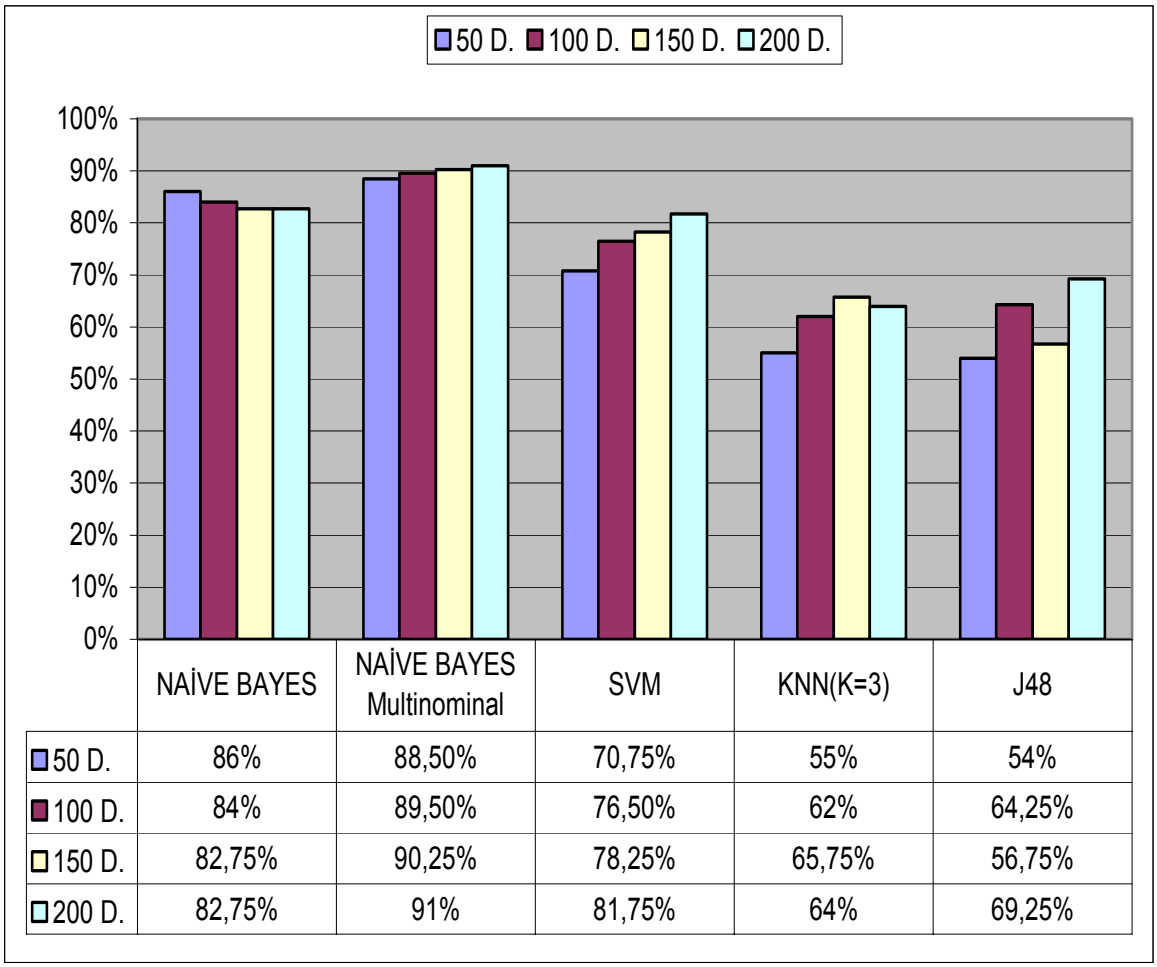
3.3. Eğitim Dokümanı Sayısının Performansa Etkisi (Bit Frekansı)

Doküman sayısının farklı algoritmalar üzerinde performansa olan ektisini değerlendirmek için, farklı sayıda eğitim setleri kullanılarak dokuman sayısının performansa etkisi incelenmiştir. Bu kısımda, eğitim seti, 50, 100, 150 ve 200 dokümanlık setlerle test edilmiş, bit frekansı ağırlığına göre, 400 dokümanlık test seti için, farklı sınıflandırma algoritmaları üzerinde doküman sayısının, performansa etkisi incelenmiştir. Tablo 5. de bit frekansına göre algoritmaların doküman sayısına bağlı başarı değerleri verilmiştir.

Tablo 5. Doküman sayısına göre bit frekansı için algoritmaların başarı değerleri

Algoritmalar	50 D.	100 D.	150 D.	200 D.	ORT.
NAİVE BAYES	%86	%84	%82,75	%82,75	%83,88
NAİVE BAYES Multinominal	%88,5	%89,5	%90,25	%91	%89,81
SVM	%70,75	%76,5	%78,25	%81,75	%76,81
KNN(K=3)	%55	%62	%65,75	%64	%61,69
J48 (Karar Ağacı)	%54	%64,25	%56,75	%69,25	%61,06

Bit frekansına göre, doküman sayısının performansa etkisi incelendiğinde başarı yüzdesinin her algoritma için %3 ile %5 oranında arttığını görüyoruz. Buradan doküman sayısına göre başarı performansının bit frekansına göre daha olumlu sonuç verdiğini söyleyebiliriz. Bit frekansına dayalı tüm ölçümler, kelime frekansına göre daha başarılı sonuçlar vermiştir. Bu başarı, tüm algoritmalarda benzer oranda görülmüştür. Tablo 5. e göre en başarılı algoritma yine Naive Bayes Multinominal modeli olmuştur. Ancak en fazla artış KNN algoritmasında yaşanmıştır. KNN algoritmasında kelime frekansına göre ortalama başarı %42 iken, aynı algoritmanın bit frekansına göre ortalama başarısı % 62' ye çıkmıştır. Yaklaşık %20 'lik bir oranla en yüksek performans artışı gözlenmiştir. Sonuç olarak, dokuman sayına göre hem kelime frekansı hem de bit frekansı için "en başarılı algoritma Naive Bayes Multinominal model dir" diyebiliriz.



	NAİVE BAYES	NAİVE BAYES Multinominal	SVM	KNN(K=3)	J48
50 D.	86%	88,50%	70,75%	55%	54%
100 D.	84%	89,50%	76,50%	62%	64,25%
150 D.	82,75%	90,25%	78,25%	65,75%	56,75%
200 D.	82,75%	91%	81,75%	64%	69,25%

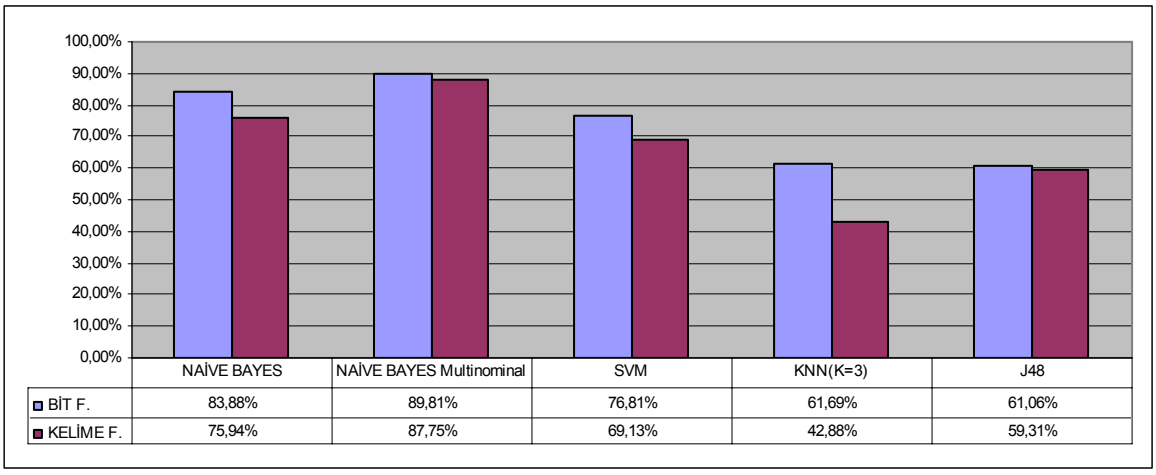
Şekil 24. Doküman sayısına göre bit frekansı için algoritmaların başarı grafiği

Eğitim dokümanı sayısına göre Şekil 24. de gösterilen grafiği açıklayacak olursak, kelime frekansında olduğu gibi, bit frekansına göre yapılan sınıflandırmada da eğitim dokümanı sayısı arttıkça tüm algoritmaların başarısı gözle görülür oranda artmıştır. Bunun yanında algoritmaların başarı yüzdesi olarak bakıldığında, sınıflandırma başarısı en yüksek algoritma, yine Naive Bayes Multinominal modeldir. Çok yüksek oranda diğer algoritmalarla performans farkı vardır. Ortalama başarısı %90' a yaklaşırken diğer algoritmalar daha düşük seviyede kalmıştır. Bu sonuçlar, Şekil 24. de değerler tablosu ile birlikte verilmiştir. Ayrıca 200 eğitim, 400 test dokümanı için bit frekansına göre, algoritmaların başarı değerleri ve sınıflara göre hata matrisleri tablosu Tablo 6. da verilmiştir. Kategorilerle ilgili sonuç ve tespitlere bir sonraki bölümde değinilecektir.

Tablo 6. Bit frekansı için sınıflara göre hata matrisleri

ALGORİTMA	BAŞARI	HATA MATRİSİ
NAİVE BAYES	**%82,75 (331/400)**	a b c d e <-- classified as 71 0 1 8 0 | a = spor 2 71 4 1 2 | b = ekonomi 0 8 72 0 0 | c = siyaset 1 2 3 **74** 0 | d = saglik 4 21 5 7 43 | e = bilisim
NAİVE BAYES Multinominal	**%89,25 (357/400)**	a b c d e <-- classified as 72 0 1 7 0 | a = spor 2 71 3 1 3 | b = ekonomi 0 8 72 0 0 | c = siyaset 1 2 2 **75** 0 | d = saglik 0 2 3 1 74 | e = bilisim
SVM	**%81,75 (327/400)**	a b c d e <-- classified as 72 2 1 5 0 | a = spor 1 71 6 1 1 | b = ekonomi 1 14 65 0 0 | c = siyaset 1 3 1 **75** 0 | d = saglik 1 27 2 6 44 | e = bilisim
KNN(K=3)	**%64 (256/400)**	a b c d e <-- classified as 68 6 0 6 0 | a = spor 5 68 6 1 0 | b = ekonomi 12 31 36 1 0 | c = siyaset 4 4 0 **72** 0 | d = saglik 4 52 1 11 12 | e = bilisim
J48 (Karar Ağacı)	**%69,25 (277/400)**	a b c d e <-- classified as **67** 8 2 3 0 | a = spor 2 50 17 8 3 | b = ekonomi 3 24 52 0 1 | c = siyaset 2 8 2 63 5 | d = saglik 0 21 10 4 45 | e = bilisim

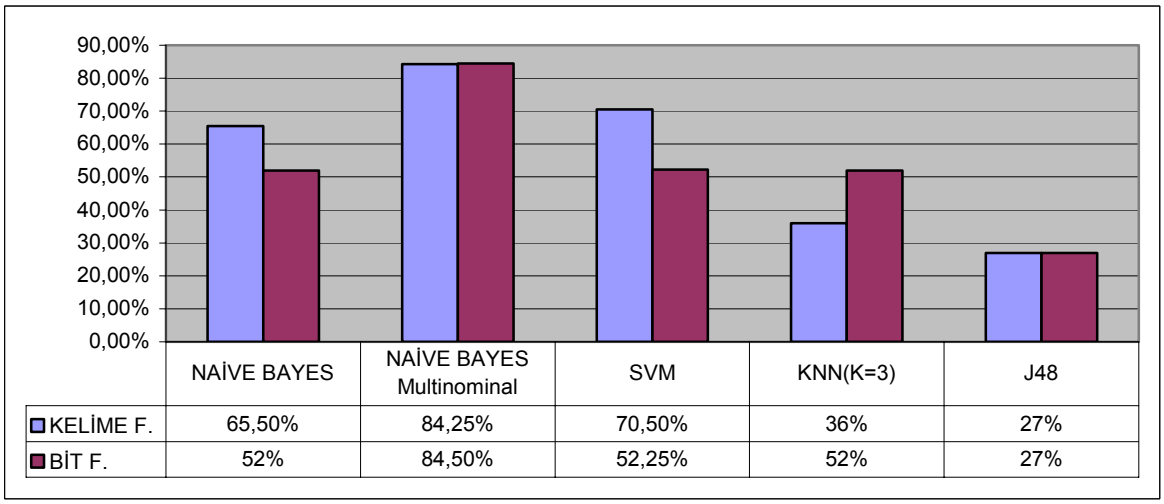
Kelime frekansı ve bit frekansına göre, değişik sayılarda (50,100,150,200) eğitim dokümanı ve 400 test dokümanı için algoritmaların başarı ölçümleri yapılmış ve sonuçlar Tablo 3. ve Tablo5. te verilmişti. Bu ölçümler sonucunda algoritmaların ortalama başarısını verecek olursak aşağıdaki grafiği elde ederiz.



	NAİVE BAYES	NAİVE BAYES Multinominal	SVM	KNN(K=3)	J48
BİT F.	83,88%	89,81%	76,81%	61,69%	61,06%
KELİME F.	75,94%	87,75%	69,13%	42,88%	59,31%

Şekil 25. Bit ve kelime frekansı için algoritmaların ortalama başarı değerleri

Doküman sayısının etkisini daha da pekiştirmek amacıyla, son olarak sistem, sadece sözlükteki kelimeleri içeren ve her kategoriden sadece 1 dokümanla eğitilip (toplam 5 eğitim dokümanı), 400 test dokümanı ile test edilmiş ve Şekil 26. daki sonuçlar elde edilmiştir. Algoritmaların başarı performansı büyük ölçüde azalmış ve dokuman sayısının performansa etkisi bir kez daha sonuçlarda görülmüştür.



	NAİVE BAYES	NAİVE BAYES Multinominal	SVM	KNN(K=3)	J48
KELİME F.	65,50%	84,25%	70,50%	36%	27%
BİT F.	52%	84,50%	52,25%	52%	27%

Şekil 26. Eğitim seti 5 doküman için algoritmaların başarı değerleri

3.4. Çapraz Doğrulamanın Algoritma Performanslarına Etkisi

Bu yöntemde eğitim seti ve test seti bir bütün olarak ele alınır. Farklı modellerle alt kümelere ayrılarak doğrulama yapılır. Doğrulama modelleri Basit Doğrulama, Çapraz Doğrulama ve N Kere Çapraz Doğrulama modelleridir.

Basit Doğrulama (Simple Validation)

Büyük veri setleri için kullanılır. Verilerin %5 ila %33 lük kısmı test seti olarak ayrılır, geri kalanıyla (training set) model kurulur. Ve daha sonra kurulan model, test seti ile test edilir ve modelin doğruluk oranı hesaplanır. Eğer farklı eğitim seti ve test verileri kullanılmazsa modelin geçerliliği tahmin üstü (over estimate) olur.

Çapraz Doğrulama (Cross Validation)

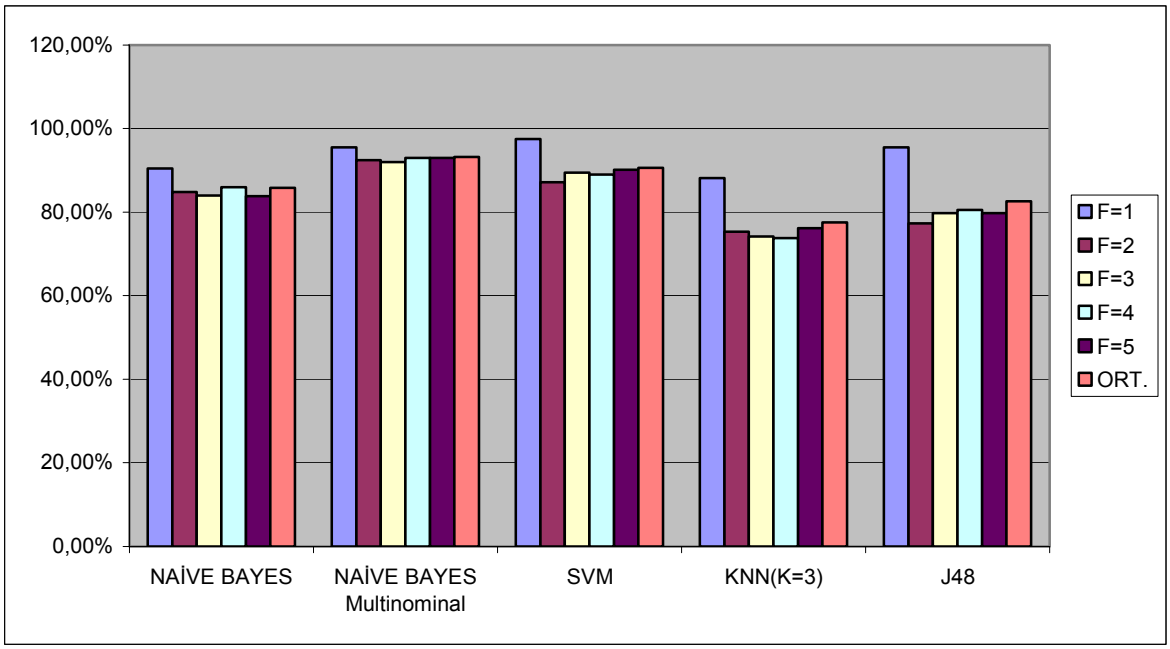
Çapraz doğrulama tüm veriyi kullanmayı sağlayan bir metottur. Veri seti rasgele 2 eşit kısma ayrılır. İlk veri setiyle sistem eğitilir, eğitilen sistem diğer veri setiyle test edilir ve doğruluk oranı hesaplanır. Daha sonra test veri setiyle sistem eğitilir ve eğitilen sistem birinci veri setiyle tahmin edilip, doğruluk oranı hesaplanır. Ve en sonunda tüm veriler kullanılarak model oluşturulur. Daha önceden hesaplanmış olan 2 doğruluk oranının ortalamasıyla, son model karşılaştırılır. n kere çapraz doğrulama (n Fold Cross Validation) da ise veri seti rasgele n gruba ayrılır. 1. grup test için ayrılırken geriye kalan n-1 grupla sistem eğitilir. Eğitilen sistem, test için ayrılan veriler üzerinde test edilir ve doğruluk oranı hesaplanır. Süreç n defa tekrar eder ve modelin doğruluk oranı, n tane doğruluk oranının ortalaması kadar olur.

Sistem, 600 test dokümanı için, 5 farklı çapraz doğrulama değeri için denenmiş ve algoritmaların performansı, kelime ve bit frekansları için ayrı ayrı ölçülmüştür.

Tablo 7. Çapraz doğrulamaya göre algoritma performansları (kelime frekansı)

	F=1	F=2	F=3	F=4	F=5	ORT.
NAİVE BAYES	%90,5	%84,8	%84	%86	%83,8	%85,82
NAİVE BAYES Multinominal	%95,5	%92,5	%92	%93	%93	%93,20
SVM	%97,5	%87,16	%89,5	%89	%90,16	%90,66
KNN(K=3)	%88,16	%75,3	%74,16	%73,8	%76,16	%77,52
J48 (Karar Ağacı)	%95,5	%77,3	%79,8	%80,5	%79,8	%82,58

Tablo 7. de verildiği gibi kelime frekansına göre yapılan çapraz doğrulama için elde edilen sonuçlara bakılırsa farklı F çapraz değerleri için sonuçlar küçük oranda değişse de performans açısından Naive Bayes Multinominal modeli, en başarılı algoritmadır. En düşük başarıya sahip algoritma ise KNN algoritmasıdır. Bu ölçümler, algoritma performansları ve ortalama değerlerle birlikte Tablo 7. de ve Şekil 27. de verilmiştir.

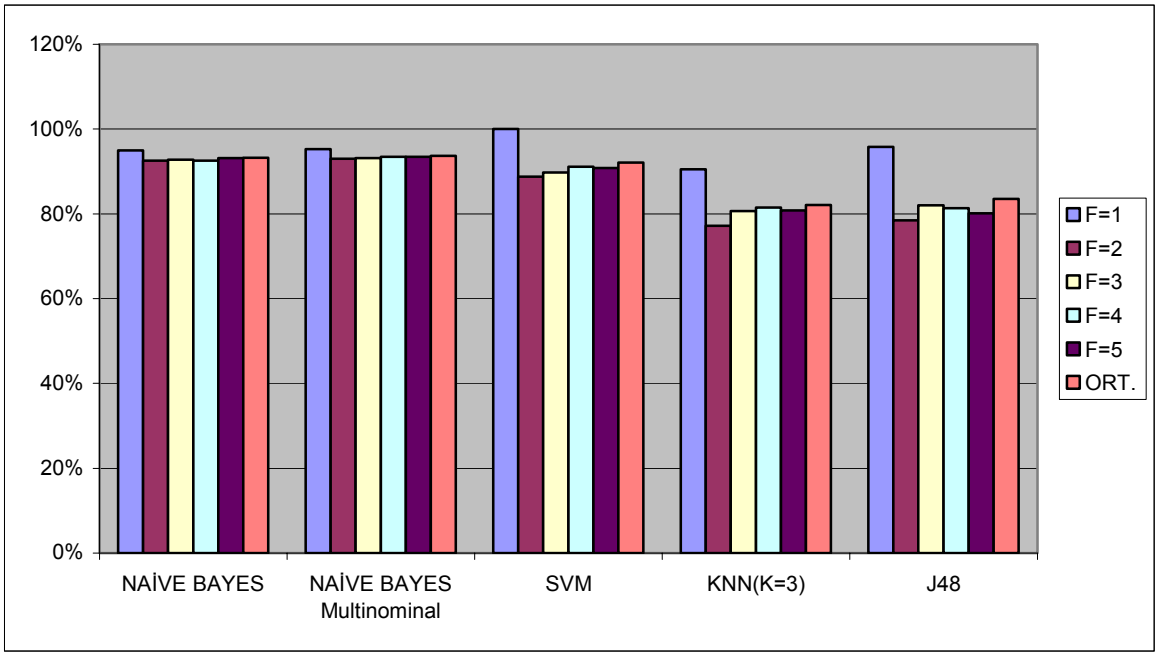


Şekil 27. Çapraz doğrulamaya göre algoritma performansları (kelime frekansı)

Bit frekansına göre yapılan çapraz doğrulama için elde edilen sonuçlara bakılırsa farklı F çapraz değerleri için, performans açısından Naive Bayes Multinominal modeli, en başarılı algoritmadır. En düşük başarıya sahip algoritma ise KNN algoritmasıdır. Bu ölçümler, algoritma performansları ve ortalama değerlerle birlikte Tablo 8. de ve Şekil 28. de verilmiştir.

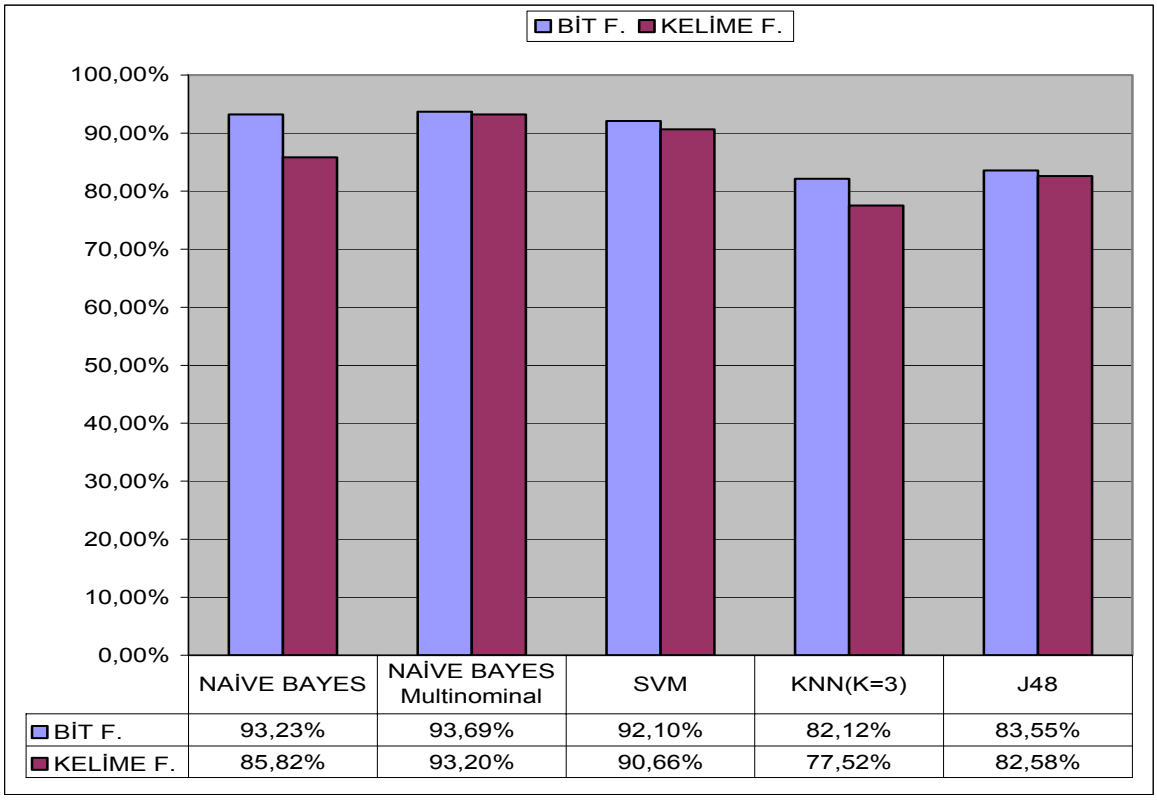
Tablo 8. Çapraz doğrulamaya göre algoritma performansları (bit frekansı)

	F=1	F=2	F=3	F=4	F=5	ORT.
NAİVE BAYES	%95	%92,6	%92,8	%92,6	%93,16	%93,23
NAİVE BAYES Multinominal	%95,3	%93	%93,16	%93,5	%93,5	%93,69
SVM	%100	%88,8	%89,8	%91,1	%90,8	%92,10
KNN(K=3)	%90,5	%77,16	%80,66	%81,5	%80,8	%82,12
J48 (Karar Ağacı)	%95,8	%78,5	%82	%81,3	%80,16	%83,55



Şekil 28. Çapraz doğrulamaya göre algoritma performansları (bit frekansı)

Çapraz doğrulamalar için, algoritma performanslarını daha iyi analiz edebilmek için Bit ve Kelime frekanslarına göre algoritma performanslarının ortalamaları alınmış ve Şekil 29. da verilmiştir.



	NAİVE BAYES	NAİVE BAYES Multinominal	SVM	KNN(K=3)	J48
BİT F.	93,23%	93,69%	92,10%	82,12%	83,55%
KELİME F.	85,82%	93,20%	90,66%	77,52%	82,58%

Şekil 29. Çapraz doğrulamalar için ortalama algoritma performansları

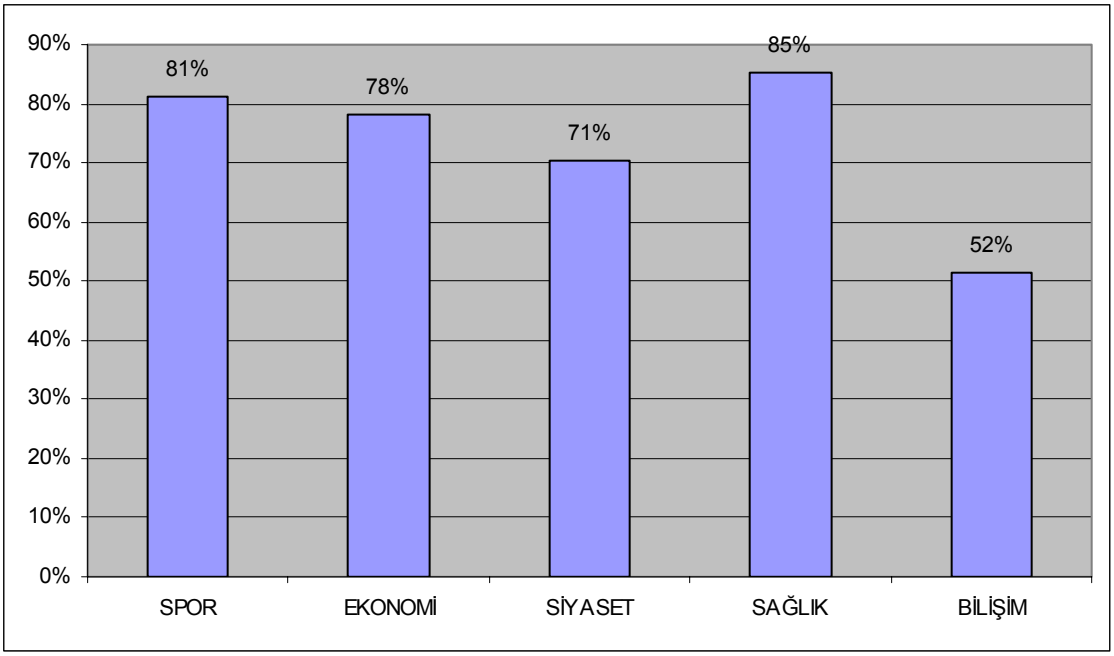
Bu sonuçlara dayanarak bit frekansının çapraz doğrulama da daha etkin bir ağırlıklandırma olduğunu söyleyebiliriz. Aradaki fark küçük de olsa bit frekansı, kelime frekansına göre daha iyi sonuç vermiştir.

3.5. Doküman Sayısına Göre Kategori Performansı

Bu kısımda 200 eğitim, 400 test dokümanı için kelime frekansına göre kategori bazında algoritmaların başarısı ölçülmüştür. Elde edilen sonuçlara göre en başarılı kategori "sağlık" kategorisi olmuştur(%85). En başarısız kategori ise "bilişim" kategorisidir(%52). Bu değerlerde eğitim dokümanlarının kalitesi büyük rol oynamaktadır. Buradan bilişim kategorisindeki eğitim dokümanlarının çok fazla ayırt edici olmadığını söyleyebiliriz.

Tablo 9. 200 eğitim 400 test dokümanı için kategori performansları (kelime f.)

	SPOR	EKONOMİ	SİYASET	SAĞLIK	BİLİŞİM
NAİVE BAYES	76/80	54/80	59/80	74/80	53/80
NAİVE BAYES Multinominal	72/80	71/80	73/80	76/80	65/80
SVM	68/80	69/80	51/80	74/80	40/80
KNN(K=3)	51/80	62/80	45/80	59/80	7/80
J48 (Karar Ağacı)	58/80	56/80	54/80	58/80	41/80
ORT.	%81	%78	%71	%85	%52

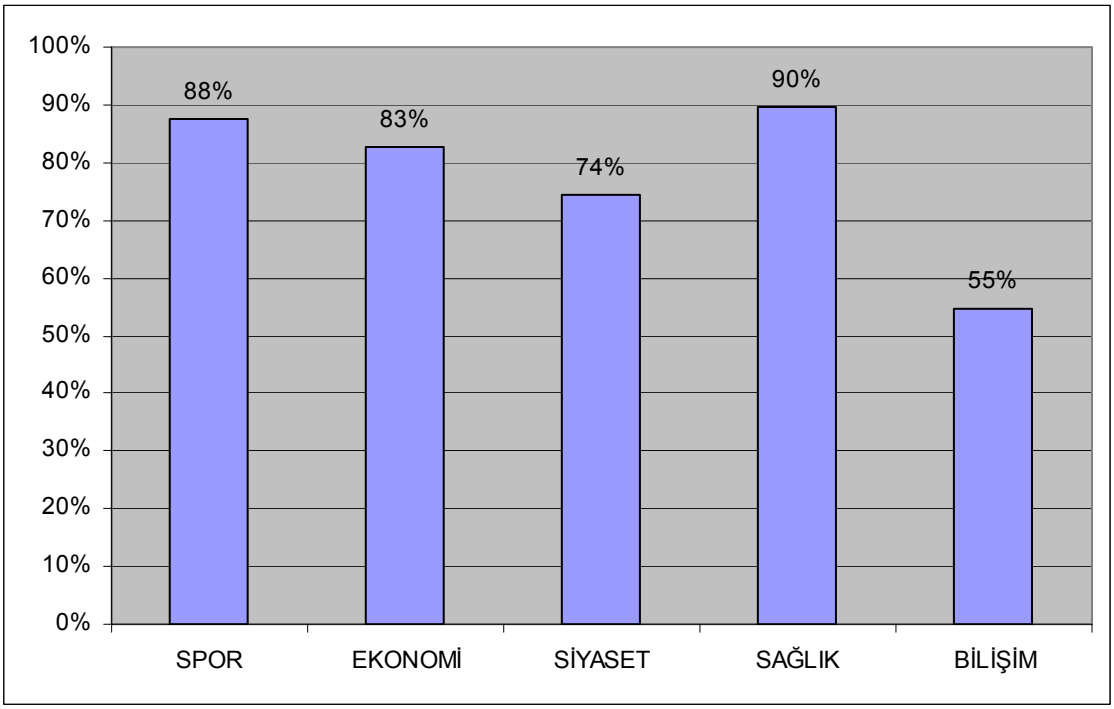


Şekil 30. 200 eğitim 400 test dokümanı için kategori performansları (kelime f.)

Bir önceki tabloda (Tablo 9.) kelime frekansı için yapılan ölçümleri yine aynı eğitim ve test verileri için bit frekansına göre modellenirse Tablo 10. da gösterilen sonuçlar elde edilir. Buna göre yine en başarılı kategori "sağlık", en başarısız ise "bilişim" kategorisidir. Bit frekansına göre kategori bazında performans, kelime frekansına göre daha da arttığı görülmüştür. Sağlık %85'ten %90'a, bilişim ise %52' den %55' e çıkmıştır.

Tablo 10. 200 eğitim 400 test dokümanı için kategori performansları (bit f.)

	SPOR	EKONOMİ	SİYASET	SAĞLIK	BİLİŞİM
NAİVE BAYES	71/80	71/80	72/80	74/80	43/80
NAİVE BAYES Multinominal	72/80	71/80	72/80	75/80	74/80
SVM	72/80	71/80	65/80	75/80	44/80
KNN(K=3)	68/80	68/80	36/80	72/80	12/80
J48 (Karar Ağacı)	67/80	50/80	52/80	63/80	45/80
ORT.	%88	%83	%74	%90	%55



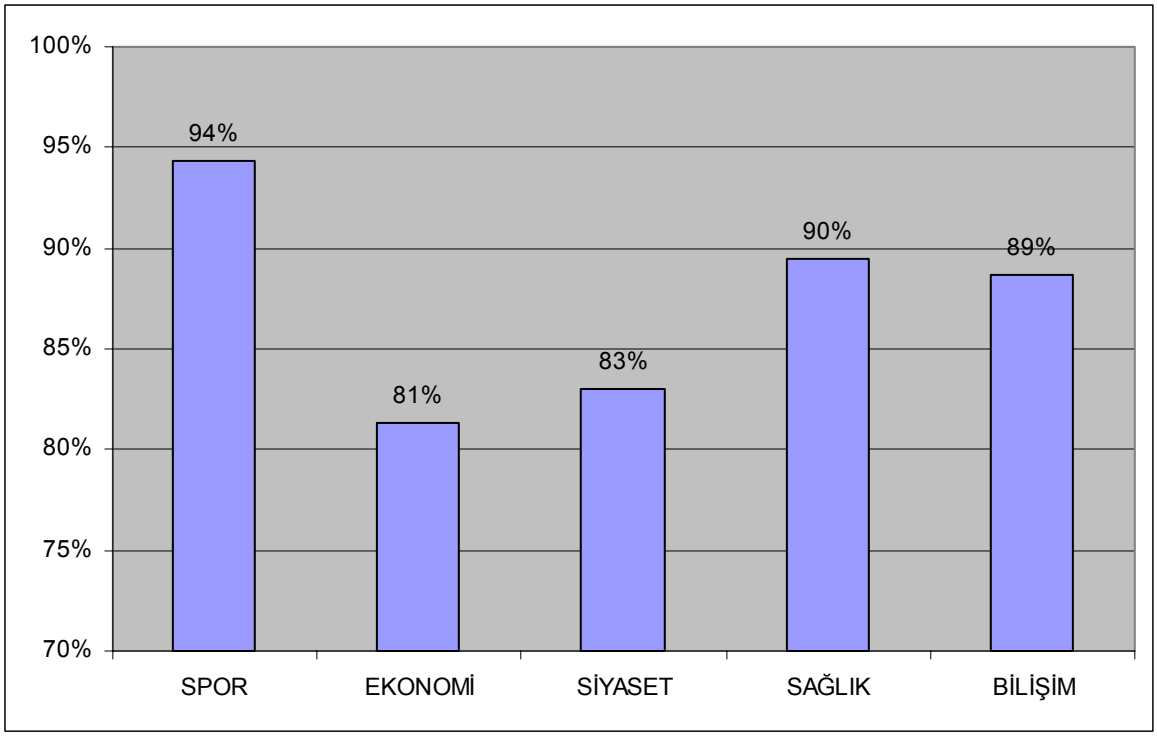
Şekil 31. 200 eğitim 400 test dokümanı için kategori performansları (bit f.)

3.6. Çapraz Doğrulamaya Göre Kategori Performansı

Çapraz doğrulama için 600 dokümanlık test verisi kullanılmıştır. Bu test seti için 1'den 5'e kadar çapraz doğrulama uygulanmıştır. Algoritmaların hata matrislerinden faydalanılarak her kategori için ortalama değerler hesaplanmıştır. Bu değerler Tablo 11. ve Şekil 32. de verilmiştir. Tablo 11. deki değerlere bakarak kelime frekansına dayalı çapraz doğrulama için en başarılı kategori "spor" kategorisidir.

Tablo 11. Çapraz doğrulama için kategori performans değerleri (kelime f.)

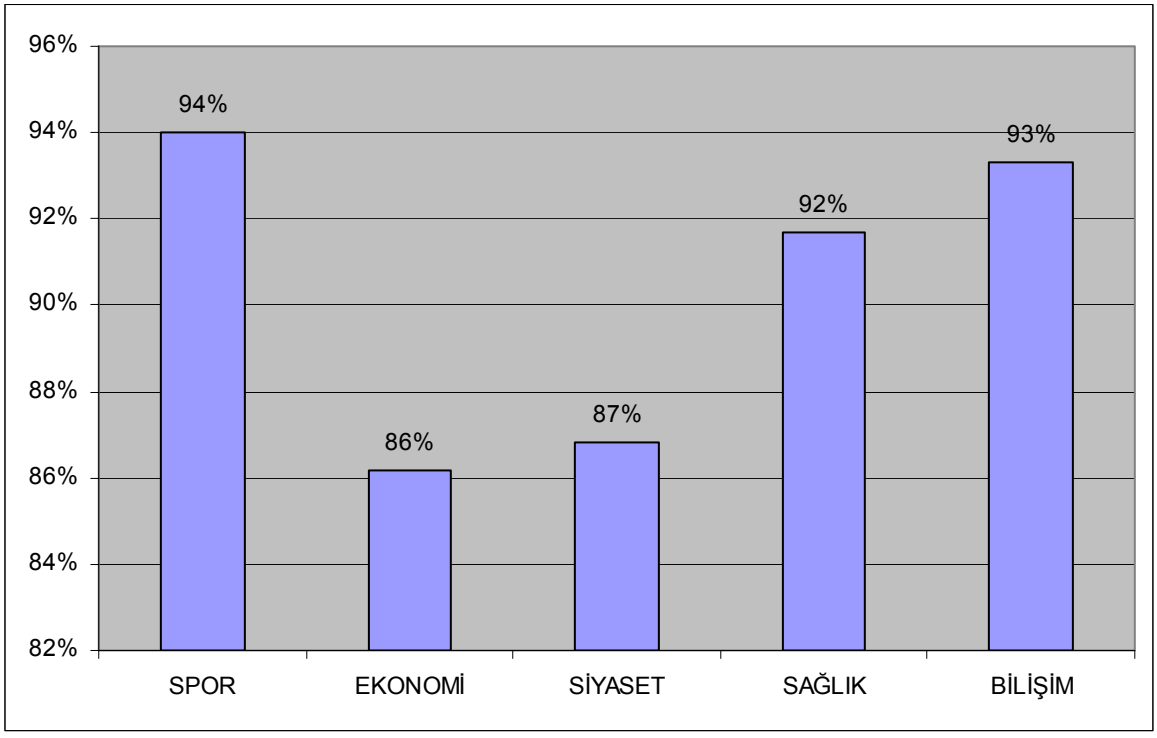
	N.B	**N.B. MULT.**	**SVM**	**KNN**	**J48**	**ORT(%)**
SPOR	113/120	113/120	113/120	113/120	114/120	94%
EKONOMİ	97/120	107/120	99/120	93/120	92/120	81%
SİYASET	103/120	113/120	105/120	84/120	93/120	83%
SAĞLIK	112/120	117/120	114/120	84/120	110/120	90%
BİLİŞİM	94/120	112/120	116/120	102/120	108/120	89%



Şekil 32. Çapraz doğrulama için kategori performans değerleri (kelime f.)

Tablo 12. Çapraz doğrulama için kategori performans değerleri (bit f.)

	N.B	**N.B. MULT.**	**SVM**	**KNN**	**J48**	**ORT(%)**
SPOR	112/120	108/120	113/120	115/120	116/120	94%
EKONOMİ	110/120	109/120	109/120	97/120	92/120	86%
SİYASET	112/120	113/120	109/120	93/120	94/120	87%
SAĞLIK	116/120	117/120	114/120	93/120	110/120	92%
BİLİŞİM	114/120	117/120	117/120	103/120	109/120	93%



Şekil 33. Çapraz doğrulama için kategori performans değerleri (bit f.)

Çapraz doğrulamaya bağlı ölçümleri toplu olarak yorumlayacak olursak, hem kelime frekansı hem de bit frekansı için en başarılı kategori "spor" kategorisi çıkmıştır. İkinci olarak "bilişim" ve "sağlık" kategorileri gelmektedir. Burada "spor" kategorisine ait terimlerin daha özel oluşu ve diğer sınıflarla ortak kelime içermemesi spor kategorisi için en önemli başarı etkenidir. Eğitim dokümanına dayalı kategori ölçümlerinde en başarısız kategori olan "bilişim" kategorisi, çapraz doğrulamada spordan sonra en başarılı kategori

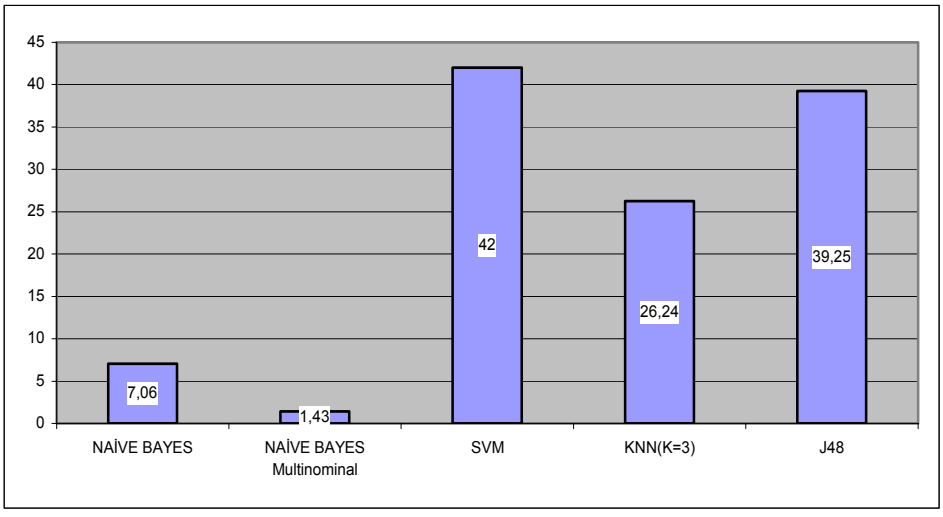
olmuştur. Bu sonuç ilk hipotezimizi doğrular niteliktedir. Bilişim ile ilgili eğitim dokümanlarının kaliteli olmadığı, bu sonuçlar la bir kez daha doğrulanmıştır. Çapraz doğrulama ile bilişim kategorisindeki başarı %55' ten %92' ye çıkarak büyük bir artış göstermiştir. En düşük orana sahip olan "ekonomi" ve "siyaset" kategorileri için de şunları söyleyebiliriz. Bu iki kategori birbirine çok yakın olduğu ve çok sayıda ortak kelime içerdiği için başarı yüzdesi ekonomi ve siyaset kategorileri için düşük çıkmıştır.

3.7. Algoritmaların Çalışma Süresine Göre Performansı

Algoritmaların çalışma sürelerini ölçmek için 1200 test dokümanı K=10 değeri için, 10 kez çapraz doğrulamaya tabi tutulmuş ve her algoritma için çalışma süresi ölçülmüştür. Çalışma süresine göre performanslar karşılaştırıldığında sınıflandırma da en başarılı algoritma olan Naive Bayes Multinominal Modeli, çalışma süresi olarak ta büyük bir farkla en başarılı algoritma olmuştur.

Tablo 13. Algoritmaların çalışma süresine göre performansı

ALGORİTMA	**SÜRE (Sn)**
NAİVE BAYES	7,06
NAİVE BAYES Multinominal	1,43
SVM	42
KNN(K=3)	26,24
J48 (Karar Ağacı)	39,25



Şekil 34. Algoritmaların çalışma süresine göre performansı

4. ÖNERİLER

Sonuç olarak, yapılan ölçümleri ve elde edilen değerleri genel olarak değerlendirecek olursak, sınıflandırma performansı açısından en başarılı metin sınıflandırma algoritması, yüksek doğruluk oranıyla, Naive Bayes Multinominal modeldir. Aynı algoritma, çalışma süreleri açısından da en başarılı algoritma olmuştur. Naive Bayes Algoritması, kolay uygulanabilirliği ve yüksek performansı ile en etkili metin sınıflandırma algoritmalarından birisidir. Naive Bayes modelinde her değişkenin, yani her kelimenin tekrar sayısının birbirinden bağımsız oluşu ve her kelime için olasılık hesabının yapılması algoritmanın sınıflandırma başarısında önemli etkendir. Çalışma süresi olarak baktığımızda, Naive Bayes modelinde, dokümanlar kosinüs benzerliği açısından birbiriyle karşılaştırılmaz. Her doküman atandığı kategoriye yakınlığı ölçülür. Bundan dolayı çalışma süresi düşüktür. Zaman performansı açısından en başarılı algoritmadır. Naive Bayes' ten sonra en başarılı sınıflandırıcı, SVM (Vektör Destek Makinası) algoritması olmuştur. Yapılan ölçümlerde K-NN (K-En Yakın Komşu) algoritması ise en düşük performansa sahip algoritma olmuştur.

Kategori performanslarını değerlendirecek olursak, doküman sayısına göre yapılan ölçümlerde sağlık kategorisi en yüksek, bilişim kategorisi ise en düşük oranda doğruluğa sahip kategoriler olmuştur. Bunda eğitim ve test dokümanlarının kalitesi büyük rol oynamaktadır. Bu ölçümleri desteklemek için yapılan çapraz doğrulamaya göre kategori performansları ölçülmüştür. Çapraz doğrulamaya göre yapılan kategori ölçümlerinde, spor ve bilişim kategorileri, en başarılı kategoriler olmuştur. Bunda spor ve bilişim kategorilerinin diğer sınıflara uzak oluşu ve daha özel anahtar kelimeler içermesi bu kategorilerin başarısını artırmıştır. Bilişim kategorisindeki yüksek başarı artışı ise tamamen eğitim dokümanlarının kalitesi ile ilişkilendirilmiştir. Bu tespit, çapraz doğrulama da çok açık bir şekilde görülmüştür. Ölçümlerde siyaset ve ekonomi kategorileri birbirine çok yakın sınıflar olduğu için kategori bazında düşük performans göstermişlerdir.

Arama sonuç madenciliği kısmında yapılan çalışma da, modellemeye çalıştığımız google tabanlı arama-sonuç analizi modülü, değişik kategoriler için test edilmiş ve doğruluğu yüksek sonuçlar alınmıştır. Arama motorları ile paralel çalışan ve özellik seçici daha çok anahtar kelime içeren bir otomatik konu sınıflandırma sistemi, aramada daha

isabetli sonuçlara ulaşmamızı sağlayacaktır. Buradan hareketle daha etkin bir veri sınıflandırma modeli ile arama - sonuç ilişkisinde daha yüksek başarı sağlanacağı görülmüştür. Otomatik sınıflandırma sistemlerinin gelişimi, gelecekte internet kullanıcılarının, doğru bilgiye erişim süreçlerine olumlu katkılar yapacaktır.

5. KAYNAKLAR

1. Etzioni, O. , The World Wide Web: Quagmire or Gold Mine, Communications of the ACM, 39-11 (1996) 65-68.

2. Amasyalı, M.F. ve Yıldırım, T., Otomatik Haber Metinleri Sınıflandırma, SIU, 2004.

3. Eyheralendy, S., Lewis, D. ve Madigan D., On the Naive Bayes Model for Text Categorization, 2003.

4. Alpaydın, E., Zeki Veri Madenciliği, Bilişim 2000 Eğitim Semineri, İstanbul, 2000.

5. Pilavcılar, İ.F., Metin Madenciliği ile Metin Sınıflandırma, Yüksek Lisans Tezi, Y.T.Ü., Fen Bilimleri Enstitüsü, İstanbul, 2007.

6. Kesgin, F., Türkçe Metinler için Konu Belirleme Sistemi, Yüksek Lisans Tezi, İ.T.Ü., Fen Bilimleri Enstitüsü, İstanbul, 2006.

7. Vahaplar, A. ve İnceoğlu M.M., Veri Madenciliği ve Elektronik Ticaret, http://inet-tr.org.tr/inetconf7/eposter/inceoglu.doc , 27 Mayıs 2009.

8. Takçı, H. ve Soğukpınar, İ., Kütüphane Kullanıcılarının Erişim Örüntülerinin Keşfi, Bilgi Dünyası, 3, 1 (2002) 12-26.

9. Diri, B. ve Amasyalı, M.F., Automatic Author Detection for Turkish Texts, Artificial Neural Networks and Neural Information Processing , (2003) 138-141

10. Amasyalı M.F. ve Diri B., Automatic Written Turkish Text Categorization in Terms of Author, Genre and Gender, 11th International Conference on Applications of Natural Language to Information Systems, Austria, 2006.

11. Takçı, H. ve Soğukpınar, İ., Classification Based Intrusion Detection, The 3rd Asia Pasific International Symposium on Information Technology, Ocak 2004, İstanbul.

12. Takçı, H. ve Soğukpınar, İ., Kullanıcı Erişim Desenleriyle Saldırı Tespiti, Bilgi Teknolojileri Kongresi, Mayıs 2002, Denizli.

13. Tatlıdil, H., Uygulamalı Çok Değişkenli İstatistiksel Analiz, Cem Web Ofset, Ankara, 1996, 352-362.

14. Apte, C., Damerau, F. ve Weiss, S.M., Automated learning of decision rules for text categorization, ACM Transactions on Information Systems, 12 (1994) 233–251.

15. Levenberg, K., A method for the solution of certain nonlinear problems in least squares, Quarterly of Applied Mathematics, 2 (1944) 164-168.

16. Dumais, S., Platt, J. ve David Heckerman, D., Inductive Learning Algorithms and Representations for Text Categorization, Proceedings of ACM-CIKM, 98, 1 (1998) 148-155.

17. Gündüz, Ş. ve Adalı, E., Web kullanıcılarının davranışları için örüntü bulma ve modelleme, İTÜ Dergisi, 3, 6 (2004) 15-24.

18. Eyheralendy, S., Lewis, D. ve Madigan D., On the Naive Bayes Model for Text Categorization, 2003.

19. Jun, H. ve Hokuan, H., An algorithm for text categorization with SVM , TENCON '02 Proceedings, 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering, 1 (2002) 47-50.

20. Ayaz, R., Web Madenciliğine bir bakış, http://www.recepayaz.com/library/web_madenciligine_birbakis.doc , 15 Mayıs 2009.

21. Simoudis, E., Reality check for data mining, IEEE Expert: Intelligent Systems and Their Applications, 11, 5 (1996) 26-33.

22. Recber, E., Sınıflandırma, http://www.emrerecber.com/?tag=siniflandirma , 12 Mayıs 2009.

23. Gündüz, Ş., Veri Madenciliği, www3.itu.edu.tr/~sgunduz/courses/verimaden/slides/d3.pdf , 10 Nisan 2009

24. Şeker, Ş.E., SVM (Support Vector Machine, Destekçi Vektör Makinesi), http://www.bilgisayarkavramlari.com/2008/12/01/svm-support-vector-machine-destekci-vektor-makinesi/ , 15 Nisan 2009.

25. Chirita, P.A., Nejdl, W., Paiu, R. and Kohlschütter, C., Using ODP metadata to personalize search, In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, Brazil, (2005) 178-185.

26. Veri Madenciliği veya Bilgi Keşfi, http://www.bilgiyonetimi.org/cm/pages/mkl_gos.php?nt=538, 28 Mayıs 2009.

27. Weka System, www.cs.waikato.ac.nz/ml/weka , 2 Şubat 2009.

28. Veri Madenciliği, http://tr.wikipedia.org/wiki/Veri-madenciliği , 18 Mayıs 2009.

29. Support vector machine, http://en.wikipedia.org/wiki/Support_vector_machine , 12 Nisan 2009.

30. K-Nearest Neighbor Algorithm, http://en.wikipedia.org/wiki/K_nearest_neighbor_algorithm , 4 Nisan 2009.

31. Veri Madenciliği, http://www.arge.com/Yayinlarimiz/Makaleler/IsDunyasi/Veri_Madenciligi.aspx , 2 Mayıs 2009.

32. Text Mining, http://www.statsoft.com/textbook/sttextmin.html , 1 Mayıs 2009.

33. An Introduction to Data Mining, http://www.thearling.com/text/dmwhite/dmwhite.htm, 2 Haziran 2009.

34. Web mining , Web structure mining, http://en.wikipedia.org/wiki/Web_mining , 15 Mayıs 2009.

ÖZGEÇMİŞ

Fatih GÜRCAN, 1976 yılında Konya'da doğdu. Sırası ile Atatürk İlkokulu, Mehmet Akif Ersoy Ortaokulu ve Kulu Lisesini bitirdi. 2000 yılında Karadeniz Teknik Üniversitesi İstatistik ve Bilgisayar Bilimleri Bölümünü bitirdikten sonra, 2005 yılında Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim dalında yüksek lisans yapmaya başladı. 2001 yılından bu yana Enformatik Bölümünde Öğretim Görevlisi olarak çalışmaktadır. İyi derecede İngilizce bilmektedir.