ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI
2013-YL-018

TÜRKÇE DOKÜMANLARIN SINIFLANDIRILMASI

Rumeysa YILMAZ

Danışmanı:
Yrd. Doç. Dr. Rıfat AŞLIYAN

AYDIN

ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE
AYDIN

Matematik Anabilim Dalı Yüksek Lisans Programı öğrencisi Rumeysa YILMAZ tarafından hazırlanan Türkçe Dokümanların Sınıflandırılması başlıklı tez, 04.02.2013 tarihinde yapılan savunma sonucunda aşağıda isimleri bulunan jüri üyelerince kabul edilmiştir.

	Ünvanı, Adı Soyadı	Kurumu	İmzası
Başkan	:Yrd. Doç. Dr. Rıfat AŞLIYA	ADÜ.	.....
Üye	:Yrd. Doç. Dr. Korhan GÜNEL	ADÜ.	.....
Üye	: Yrd. Doç. Dr. Refet POLAT	Yaşar Üniv.	.....

Jüri üyeleri tarafından kabul edilen bu Yüksek Lisans tezi, Enstitü Yönetim Kurulunun ……Sayılı kararıyla **……..** tarihinde onaylanmıştır.

Prof. Dr. Cengiz ÖZARSLAN
Enstitü Müdürü

ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE
AYDIN

Bu tezde sunulan tüm bilgi ve sonuçların, bilimsel yöntemlerle yürütülen gerçek deney ve gözlemler çerçevesinde tarafımdan elde edildiğini, çalışmada bana ait olmayan tüm veri, düşünce, sonuç ve bilgilere bilimsel etik kuralların gereği olarak eksiksiz şekilde uygun atıf yaptığımı ve kaynak göstererek belirttiğimi beyan ederim.

04/02/2013

Rumeysa YILMAZ

ÖZET

TÜRKÇE DOKÜMANLARIN SINIFLANDIRILMASI

Rumeysa YILMAZ

Yüksek Lisans Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Yrd. Doç. Dr. Rıfat AŞLIYAN
2013, 75 sayfa

İnternetin hızla gelişmesi elektronik ortamdaki bilgileri ve işlemleri arttırmıştır. Fakat, bu ortamlarda depolanan ve işlenen bilgilerin boyutunun artması aranan bilgiye erişmekte problemler çıkarmıştır. Bu doğrultuda, kullanıcıların istedikleri bilgiye daha doğru ve hızlı bir şekilde ulaşma ihtiyacı doğmuştur ve elektronik ortamdaki dokümanların sınıflandırılmasında yeni metotların geliştirilmesi çalışmaları devam etmektedir. Bu çalışmada, Türkçe metin içerikli web sitelerinden elde edilen dokümanların sınıflandırılması amaçlanmaktadır.

Dokümanlar, gövde tabanlı, sözcük tabanlı, hece tabanlı ve karakter tabanlı olmak üzere dört farklı kategoride ele alınmıştır. Gövde, sözcük, hece ve karakterler için n-gram analizleri yapılmıştır. Sistem K-En Yakın Komşu Modeli (K-NN), Çok Katmanlı Algılayıcı Modeli (MLP) ve Destek Vektör Makinesi (SVM) olmak üzere üç farklı yöntem ile eğitilmiş ve test edilmiştir. Çalışmanın gerçekleştirilmesinde eğitim ve test olmak üzere iki derlem oluşturulmuştur. Her biri internet ortamından temin edilen 75'er doküman içeren eğitim, ekonomi, kültür-sanat, otomobil, sağlık ve spor sınıfları ele alınmıştır. Bu dokümanlardan 25'er tane alınarak toplamda 150 doküman sistemin eğitilmesinde, 50'şer tane alınarak toplamda 300 doküman da sistemin test edilmesinde kullanılmıştır. Çalışmada sisteme verilen dokümanlar öncelikle önişlemden geçirilmiştir. Önişlemden geçirilen dokümanların frekansları ve olasılıkları hesaplandıktan sonra her bir sınıf için öznitelik vektör veritabanı oluşturulmuştur. Öznitelik vektör veritabanı oluşturulurken sözcüklerin dokümanlarda karşılaştırılmasında eşik değeri olarak 0,25, 0,50, 0,75 ve 0,90 değerleri kullanılmış. Eğitim setindeki dokümanlar sisteme verilmiş ve her bir sınıf için oluşturulan öznitelik vektör veritabanındaki sözcükler ile karşılaştırılarak dokümanın hangi sınıfa dahil olduğu belirlenmiştir. Daha sonra test setindeki dokümanlar sisteme verilmiş ve sistemin

başarısı, kesinlik skoru, hassasiyet skoru, F-ölçüsü ve doğruluk değerlerine göre tespit edilmiştir.

Sonuç olarak en yüksek doğruluk başarı oranı SVM metodu kullanılarak sözcük 1-gramlarda %99,9 olarak bulunmuştur. F-ölçüsü değeri de %99,7 olmuştur.

Anahtar sözcükler: Doküman Sınıflandırma, K-En Yakın Komşu Modeli, Çok Katmanlı Algılayıcı Modeli, Destek Vektör Makinesi, n-gram.

ABSTRACT

TURKISH DOCUMENT CLASSIFICATION

Rumeysa YILMAZ

M.Sc. Thesis, Department of Mathematics

Supervisor: Assist. Prof. Dr. Rıfat AŞLIYAN

2013, 75 pages

Advancing the technologies of Internet has caused a great deal of digital information and operations. But, because of great amount of digital information, there are some difficulties to reach the information which is stored in databases and processed by systems. In this way, the users in information technology require to reach the information faster and more robust. For that reason, in the classification of the documents in digital technology, the studies about development of new approaches are ongoing. In this study, we aimed to the classification of the documents which are obtained from Turkish web sites.

The documents are categorized to some different classes according to word, stem, syllable and character based approaches. At the same time, n-grams of above units are also used for the classification. The developed systems classify the web based documents with the methods: K-Nearest Neighbor (K-NN), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). The systems which use MLP and SVM have been trained and tested. For training and testing operations, two corpora are generated from Turkish web pages. The documents are classified into 6 different classes as "education", "economy", "art and culture", "automobile", "health" and "sport". Each class includes 25 documents and 50 documents in training and testing sets respectively. Thus, the corpora have totally 450 documents for training and testing operations. In preprocessing stage of the system, all unnecessary characters such as punctuation marks are removed from the documents. All capital letters are converted to lower case and only one space character is allowed between two consecutive words. After the frequencies of word, stem, syllable and character n-grams in all documents have been computed in feature extraction stage, the documents have been represented as column vectors which contain frequencies. The tokens as word, stem, syllable and character n-grams are determined with threshold values as 0.25, 0.50, 0.75 and 0.90 in feature selection. In training stage, every document as a feature vector is

given to MLP and SVM methods and using these methods a model is constructed for each class. Finally, the documents in test set are categorized to the classes using the models. The designed systems are evaluated according to the Precision, Recall, Accuracy and F-measure.

The most successful method is SVM with word 1-gram and Accuracy and F-measure score of the systems are 99.9 % and 99.7 % respectively.

Key words: Document Classification, K-Nearest Neighbor, Multi-Layer Perceptron, Support Vector Machine, n-gram.

ÖNSÖZ

Eğitim hayatım boyunca maddi manevi her türlü imkanı sağlayan anneme ve babama, bana her zaman destek olan Yıldız ablama, her yönüyle takdir ettiğim ve örnek aldığım saygıdeğer hocam Yrd. Doç. Dr. Rıfat Aşlıyan'a ve hiçbir zaman yardımlarını esirgemeyen çok değerli hocam Yrd. Doç. Dr. Korhan Günel'e teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

EKLER DİZİNİ

1. GİRİŞ

Günümüzde internet ortamında ve veri tabanlarında çok büyük miktarda bilgi depolanmaktadır. Teknolojinin gelişmesiyle beraber elektronik ortamdaki dokümanların sayısı artmış ve buna bağlı olarak istenilen dokümana ulaşabilmek zorlaşmıştır. İstenilen dokümanın bu kadar büyük veriler arasından el ile seçilip çıkarılması oldukça zor ve zaman alıcı bir süreçtir. Web sayfalarının sınıflandırılması, ulaşılması istenilen bilgiye hızlı bir şekilde erişimi sağlayacaktır.

Doküman sınıflandırma çalışmaları 1960'lı yıllarda başlamıştır. Bu alanda yapılan çalışmalarla gereksiz bilgilerin kullanıcıya ulaşması engellenerek istenilen bilgiye daha hızlı ve daha doğru bir şekilde ulaşılması kolaylaşmıştır. Doküman sınıflandırmanın amacı bir dokümanın özniteliklerine bakarak önceden belirlenmiş belli sayıdaki kategorilerden hangisine dâhil olacağını belirlemektir. Bunun için çeşitli sınıflandırma yöntemleri geliştirilmiştir. Yaygın olarak kullanılan sınıflandırma yöntemleri; Naive Bayes (Kim vd., 2002), Karar Ağaçları (Wu vd., 2006) , K-En Yakın Komşu Modeli (K-NN) (Soucy ve Mineau, 2001), Maksimum Entropi Modelleri (Li vd.; Kazama ve Tsujii, 2005), Bulanık Mantık Teorisi Yaklaşımları (Liu ve Song, 2003), Destek Vektör Makineleri (Joachims, 1998; Yang ve Liu, 1999) ve Yapay Sinir Ağlarıdır (YSA).

Otomatik doküman sınıflandırma; bilgi alma, bilgi çıkarma, doküman indeksleme, doküman filtreleme, otomatik olarak meta-data elde etme ve web sayfalarını hiyerarşik olarak düzenleme gibi pek çok alanda kullanılır. Fakat Türkçe dokümanların sınıflandırılmasında bazı sorunlarla karşılaşılmıştır.

Türkçe sondan eklemeli bir dil olduğundan sözcüklere eklenen ekler ile farklı birçok sözcük elde edilebilir. Bu açıdan sözcüklerin morfolojik analizlerinin yapılmasının büyük önemi vardır. Morfolojik analiz sözcüklerin yapısının belirlenmesi işlemidir. Türkçe bir sözcüğü incelediğimizde kök ve eklerden oluştuğunu görürüz. Türkçe sözcüklerde anlamlı en küçük parça kök ve bundan sonra gelenler ise ektir. Köke gelen her bir ek sözcüğe farklı bir anlam katar. Morfolojik analiz ile sözcüklerin olası kök ve ekleri belirlenirken sözcük köklerine yapım eklerinin gelmesiyle oluşan gövdeleri de belirlenir. Örneğin; "gözlükçüden" sözcüğünün morfolojik analizini yaparsak göz-lük-çü-den olacaktır.

Sözcüklerin morfolojik analizi iki kısımda yapılır: Kök analizi, Ek analizi. Dili oluşturan sözcüklerin yapısının incelenmesi doğal dil işleme ve metin madenciliği alanları için önemli bir rol oynamaktadır. Bir dili anlamamız için o dilin özelliklerinin bilinmesi, cümle ve sözcük yapılarının belirlenmesi gerekir. Türkçe gibi eklemeli dillerde doğal dil işleme çalışmalarının yapılmasında zorluklar oluşmaktadır. Türkçe bir cümlede sözcüklerin türlerinin doğru tespit edilebilmesi için hangi sözcüğün yüklem, hangi sözcüğün özne, hangi sözcüğün nesne olduğu belirlenmelidir. Sözcük türlerinin belirlenmesi sözcüğün kök ve eklerine doğru bir şekilde ayrılmasıyla olur. Bu nedenle bu çalışmada doğru heceleme yapabilen bir sistem kullanılmıştır.

Ayrıca bazı sözcükler tek başlarına anlam taşıyabildikleri gibi yanlarına gelen başka sözcükler ile birlikte farklı anlamlar da taşıyabilirler. Böyle bir durumda sözcüklerin n-gram analizlerinin yapılması daha sağlıklı sonuçlar elde etmemizi sağlayacaktır. n-gram analizi bir dizideki tekrar oranını bulmaya yarayan bir yöntemdir. Buradaki "n" dizideki tekrar sayısını ifade eder. Bu dizi için, harf n-gramları olabildiği gibi hece n-gramları ve sözcük n-gramları da olabilir. Örneğin "otomatik doküman sınıflandırma" dizinin karakter 2-gramlarını bulalım.

ot	to	om	ma	at	ti	ik
1	1	1	3	1	1	1

do	ok	kü	üm	an
1	1	1	1	2

sı	ın	nı	ıf	fl	la	nd	dı	ır	rm
1	1	1	1	1	1	1	1	1	1

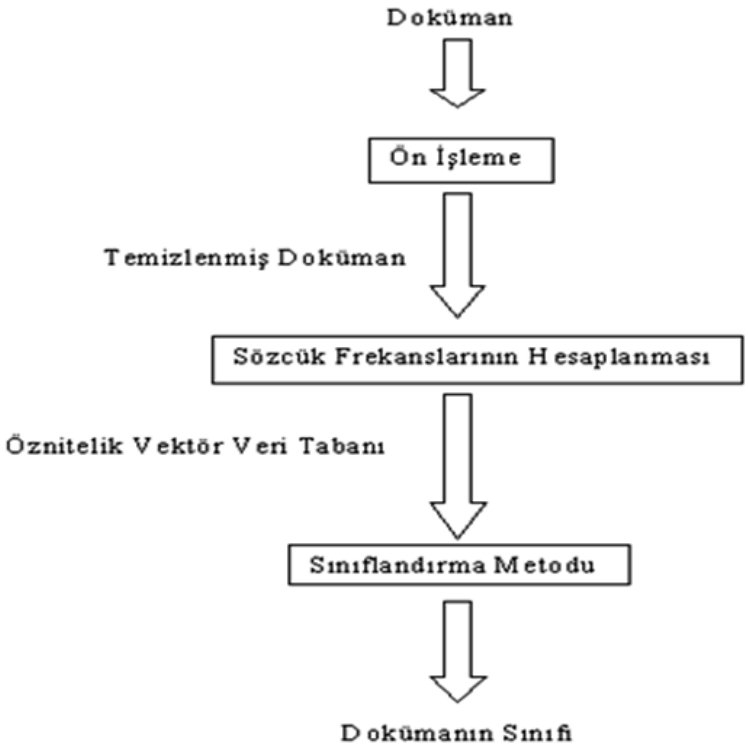
Bu çalışmada, dokümanlar dört farklı kategoriye göre incelenmiştir. Bunlardan ilkinde sözcüklerin morfolojik analizleri yapılarak elde edilen sözcük gövdelerinin 1-gram, 2-gram, ve 3-gramları oluşturulmuştur. Ayrıca dokümanlar sözcük bazında ele alınıp sözcük 1-gam, 2-gram ve 3-gramları elde edilmiştir. Bir diğer kategoride dokümanlardaki sözcükler hecelere ayrılmış olup hece 1-gram, 2-gram ve 3-gramları oluşturulmuştur. Son olarak dokümanların karakter analizi yapılmış ve karakter 2-gram, 3-gram, 4-gram, 5-gram ve 6-gramları oluşturularak sonuçlar elde edilmiştir.

Geliştirilen sistemler, Destek Vektör Makineleri, K-En Yakın Komşu Modeli ve yapay sinir ağlarından Çok Katmanlı Algılayıcı Ağı olmak üzere üç farklı yöntem kullanılarak eğitilmiştir.

Sistemin eğitimi ve test aşamasında kullanılmak üzere iki derlem oluşturulmuştur. "Eğitim", "Ekonomi", "Kültür-Sanat", "Otomobil", "Sağlık", "Spor", gibi sınıfların her biri için internet ortamından elde edilen 75 tane Türkçe doküman ele alınmış, bunlardan 25'er tanesi sistemin eğitilmesi aşamasında 50'şer tanesi ise test aşamasında kullanılmış ve sistemlerin karşılaştırılması sağlanmıştır.

Otomatik doküman sınıflandırmada sisteme verilen dokümanların sayısının önemli bir rolü vardır. Dokümanların gereğinden fazla olması sistemin öğrenmesini zorlaştırmakta, gereğinden az olması da sonuçlardaki hata oranını arttırmaktadır.

Doküman sınıflandırma Şekil 1.1'de gösterildiği gibi üç safhadan oluşmaktadır: Ön İşleme Safhası, Özniteliklerin Tespiti ve Sınıflandırma Metodu.



Şekil 1.1. Doküman sınıflandırmanın genel yapısı

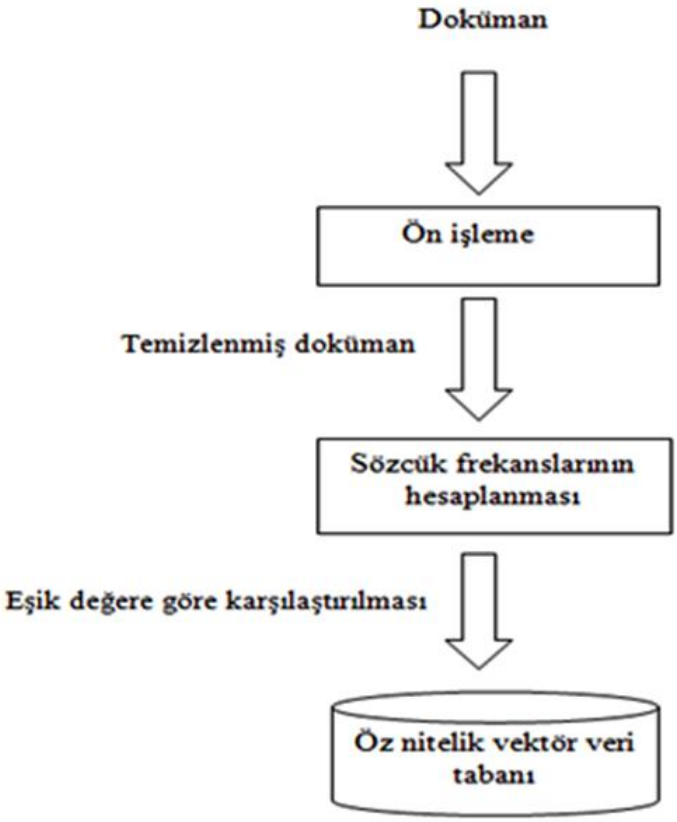
2. SİSTEM MİMARİSİ

Metin dokümanları oldukça fazla sözcük içerirler. Bazı sözcükler vardır ki bunların bütün dokümanlardaki frekansı oldukça yüksektir. Bunlara Türkçede çok sık kullanılan; "gibi", "ise", "yani", "veya", "ama", "ne", "neden", "şey", "hiç" sözcükleri örnek verilebilir. Bundan dolayı bu sözcükler ayırt edici özelliğe sahip değillerdir ve dokümanlardan elenir. Eleme işlemi indeksleme işlemi olarak adlandırılır ve bunu takip eden adımlardan oluşur.

İndeksleme işlemi sırasında dokümanlar ön işlemden geçirilirler. Önişleme safhasında dokümanlardaki boşluk, rakam ve noktalama işareti gibi herhangi bir anlam ifade etmeyen karakterler elenir, büyük harfler küçük harflere dönüştürülerek temizlenmiş doküman haline getirilir. Bu aşamadan sonra Eğitim, Sağlık, Spor, Ekonomi, Kültür-Sanat ve Otomobil kategorileri için öznitelik vektör uzayının oluşturulması gerekir.

Temizlenmiş olan bu dokümanlardaki sözcüklerin frekansları ve doküman içinde bulunma olasılıkları hesaplanır. Her bir sınıf için eğitim aşamasında kullanılacak olan dokümanlar sisteme verilerek önişlemden geçirilir. Frekansları hesaplanan bu sözcüklerin diğer dokümanlarda bulunma olasılığı belli bir eşik değerinden düşük ise bu sınıfın öznitelik vektör uzayı içerisine alınır. Böylece her sınıfı temsil edecek olan öz nitelik vektör uzayları belirlenmiş olur. Şekil 2.1.'de öznitelik vektör uzayını oluştururken izlenen adımlar verilmiştir.

Öznitelik vektör uzayı gövde tabanlı, sözcük tabanlı, hece tabanlı ve karakter tabanlı olmak üzere dört farklı kategoriye göre elde edilmiştir.



Şekil 2.1. Öznitelik vektör veri tabanının oluşturulması

Öznitelik vektör uzayları oluşturulduktan sonra internet ortamından toplanan her kategoriye ait 75 dokümandan 25 tanesi sisteme verilerek eğitilir. Test aşamasında sistemin daha önceden görmediği diğer 50 doküman sisteme verilerek üç farklı eğitim yöntemi ile sonuçlar elde edilmiştir.

3. MATERYAL VE METOT

3.1 Sınıflandırma

Farklı sınıflara ait nesneler verilmiş olsun. Yeni karşılaşılan bir nesneyi bu sınıflardan birine atama problemi sınıflandırma problemi olarak adlandırılır. İki farklı sınıfımız bulunduğunu varsayalım, bu durumda sınıflandırma problemi matematiksel olarak aşağıdaki gibi ifade edilebilir.

Farzedelim ki Denklem 3.1'deki deneysel veriler verilmiş olsun:

$$(x_1, y_1), \dots, (x_m, y_m) \in X \times \{\pm 1\} \tag{3.1}$$

Burada X, x_i örüntülerinin (ya da bir başka değiş ile gözlemlerin) alındığı boştan farklı bir kümedir. y_i ler ise etiket ya da çıktı olarak adlandırılırlar. Dikkat edilir ise bu örnekteki örüntüler $+1$ ve -1 olarak etiketlenmiş iki sınıfa aittir. Bu tip sınıflandırmalara ikili örüntü tanıma ya da ikili sınıflandırma denilmektedir.

Öğrenmede, daha önceden bilinmeyen verileri genelleştirebilmek istenmektedir. Örüntü tanıma probleminde bunun anlamı, yeni bir $x \in X$ örüntüsü verildiğinde, buna karşılık gelen $y \in \{\pm 1\}$ etiketini tahmin etmektir. Bunun için X ve $\{\pm 1\}$ üzerinde benzerlik ölçütlerine ihtiyaç vardır.

$\{\pm 1\}$ üzerinde benzerlik tanımlamak kolaydır: İki çıktı ya aynı ya da farklıdır. Gözlem kümesi üzerinde benzerliği tanımlamak için ise Denklem 3.2.'deki formdaki benzerlik ölçütünü göz önüne alalım:

$$k: X \times X \to \mathbb{R}, \quad \forall (x, x') \in X \times X \; için \; k(x, x') \in \mathbb{R} \tag{3.2}$$

Bu ölçüt, x ve x' olarak verilen iki örüntü için benzerliği karakterize eden bir gerçel sayıya döndürmektedir. Bu fonksiyon, çekirdek olarak adlandırılmaktadır.

Basit benzerlik ölçütlerinden biri noktasal çarpımdır. $\boldsymbol{x}, \boldsymbol{x'} \in \mathbb{R}^N$ olarak verilen iki vektör için noktasal çarpım Denklem 3.3'de tanımlanır:

$$(\boldsymbol{x} \cdot \boldsymbol{x'}) = \sum_{i=1}^{N} (x)_i (x')_i \tag{3.3}$$

Burada $(x)_i$, $\boldsymbol{x}$ vektörünün i. elementidir.

Noktasal çarpımın geometrik yorumu, boyları 1'e normalize edilmiş x ve x' vektörlerinin arasındaki açının kosinüsüdür. Dahası, Denklem 3.4'de gösterildiği üzere x vektörünün uzunluğunun hesaplanmasında da kullanılır:

$$\|x\| = \sqrt{(x \cdot x)} \tag{3.4}$$

İki vektör arasındaki uzaklık, fark vektörünün uzunluğu olarak hesaplanır.

Dikkat edilmelidir ki, örüntülerin bir noktasal çarpım uzayında var olduklarına dair varsayımda bulunulmamıştır. Noktasal çarpımı benzerlik ölçütü olarak kullanabilmek için örüntüleri bir noktasal çarpım uzayı $\mathcal{H}$ de vektörler olarak tanımlamak gerekmektedir. Bunun için Denklem 3.5'deki gibi dönüşüm kullanılabilir:

$$\phi: X \rightarrow \mathcal{H},\ \forall x \in X\ için\ f(x) = \mathbf{x} \tag{3.5}$$

Burada $\mathcal{H}$ uzayı, öz nitelik uzayı olarak adlandırılır.

Denklem 3.6'da gösterildiği gibi ϕ yardımı ile verileri $\mathcal{H}$ uzayına taşımak, $\mathcal{H}$ üzerinde noktasal çarpımdan bir benzerlik ölçütü tanmlamayı sağlar:

$$k(x, x') = (\boldsymbol{x} \cdot \boldsymbol{x}') = (\phi(x) \cdot \phi(x')) \tag{3.6}$$

3.2. Bazı Doküman Sınıflandırma Metotları

3.2.1. Naive Bayes

İstatistiksel sınıflandırma modelleri arasında yer alan Bayes ağları, dokümanın sınıfının belirlenmesinde sözcüklerin ve sınıfların koşullu olasılıkları kullanılır. Sınıflandırma amacı ile kurulan bir Bayes ağı kendisine yöneltilen sorulara karşılık sonuçlar üretir. Bir Bayes ağının kullanılabilmesi için olasılık modelinin oluşturulması gerekir.

Naive Bayes ağları ile otomatik doküman sınıflandırma yapılırken sistem oluşturulduktan sonra kullanılacak olan dokümanlar ve sınıflar belirlenir. Elde edilen veriler ile sözcüklerin dokümanlardaki olasılıkları hesaplanarak sisteme verilen dokümanın sınıfı tahmin edilir.

$D = \{d_1, d_2, \dots d_n\}$ Hangi sınıfa ait olduğu bilinmeyen dokümanlar olsun. $C_1, C_2, \dots C_m$ değerlerinin de sınıfları temsil ettiğini varsayalım. Bu durumda sınıfı belirlenecek olan dokümanın olasılığı Denklem 3.7'de verilmiştir.

$$P(C_i|D) = \frac{P(D|C_i)P(C_i)}{P(D)} \quad (3.7)$$

Her bir dokümanın birbirinden bağımsız olduğu kabul edilerek $P(D|C_i)$ olasılığı Denklem 3.8'deki gibi basitleştirilir (Han,2006).

$$P(D|C_i) = \prod_{k=1}^{n} P(d_k|C_i) \quad (3.8)$$

Sınıfı bilinmeyen bir dokümanın sınıfını belirlemek için Denklem 2.8'de $P(C_i|D)$ deki paydalar eşit olduğundan paylar karşılaştırılır. Denklem 3.9'da gösterildiği gibi elde edilen değerler içinden en büyük olanı seçilerek dokümanın sınıfı belirlenir.

$$arg\ max_{C_i}\{P(D|C_i)P(C_i)\} \quad (3.9)$$

Bu ifade en büyük sonrasal sınıflandırma yöntemi (Maximum A Posteriori Classification=MAP) olarak bilinir. Sonuç olarak Bayes Sınıflandırıcısı Denklem 3.10' daki gibi hesaplanır.

$$C_{MAP} = arg\ max_C \prod_{k=1}^{n} P(d_k|C_i) \quad (3.10)$$

3.2.2. Karar Ağaçları

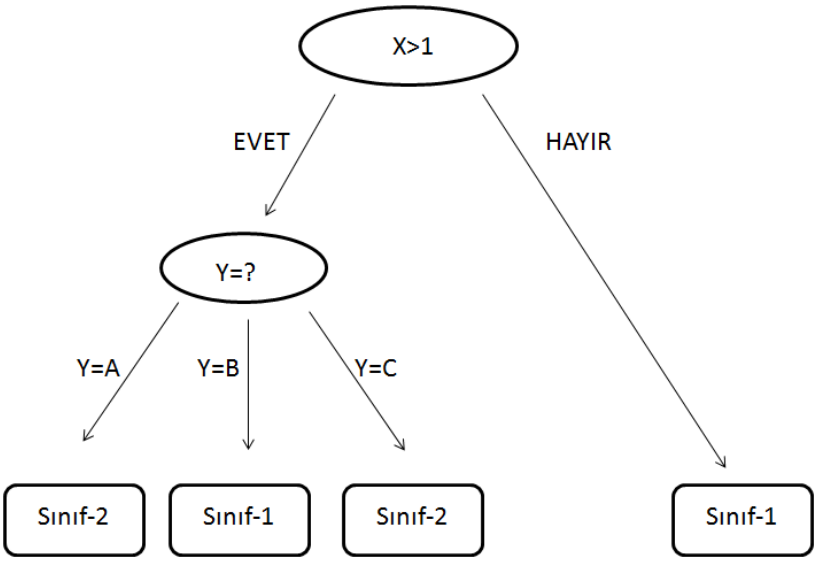
En sık kullanılan sınıflandırma yöntemlerinden biri olan karar ağaçları bir olayı sorunun cevabından yola çıkarak sonuçlandırır. Diğer sınıflandırma yöntemleriyle karşılaştırıldığında oluşturulması ve anlaşılması daha kolaydır. Bir karar ağacının yapısı düğüm, dal ve yaprak olmak üzere 3 bölümden oluşur. Ağacın düğümü soruları temsil ederken, dal soruların cevaplarını yaprak da kararın verildiği sınıfı temsil eder.

Karar ağaçlarının oluşturulmasında temel olarak kullanılan yöntemler:

- Entropiye dayalı algoritmalar
- Sınıflandırma ve regresyon ağaçları

- Bellek tabanlı sınıflandırma algoritmaları

Bir örnek sınıfı için X ve Y iki giriş niteliği olsun. $X > 1$ ve $Y = B$ koşulunu sağlayan örnekler ve $X \leq 1$ koşulunu sağlayan örnekler sınıf-1'de, $Y = A$ ve $Y = B$ koşulunu sağlayanlar ise sınıf-2'de yer alsın. Bu durumda karar ağacı Şekil 3.1'deki gibi olur (Özkan, 2008).



Şekil 3.1. Karar ağacı yapısı

3.2.3. Bulanık Mantık Teorisi Yaklaşımları

İlk olarak 1965 yılında Lotfi Zadeh tarafından ortaya atılan bulanık mantık karar verme mekanizması olarak da tanımlanabilir. Klasik mantıkta bir eleman ya bir kümeye aittir ya da değildir. Bulanık mantıkta ise bir eleman birden fazla kümeye ait olabilir. Bir doküman birden fazla sınıfa veya kategoriye dâhil olabileceğinden doküman sınıflandırmada bulanık mantık yaklaşımı kullanılabilir.

Klasik mantık kesinlik içerirken bulanık mantıkta yaklaşıklık kavramı vardır. Klasik mantıkta veriler kesin ve tam olmak zorunda olduğundan karmaşık sistemlerde bulanık mantık tercih edilir. Günlük hayatta bir nesne klasik mantığa göre evet-hayır, doğru-yanlış, soğuk-sıcak, pişmiş-çiğ, kısa-uzun olarak

sınıflandırılır bulanık mantıkta ise bunların arasındaki değerleri yani çok uzun, orta, az pişmiş, çok soğuk, çok doğru gibi yaklaşık değerleri de alır.

Bulanık mantık her ne kadar yaklaşıklık kavramı içerse de bulanık mantıkta modelleme belirsiz işlemlerle değil, kuralları ve değişkenleri esnek bir şekilde belirlenen işlemlerle yapılır. Bir bulanık modelde asıl yapı bozulmadan bulunulan duruma veya ortama göre farklı cevaplar üretilebilir (Kıyak, 2003). Bulanık mantıkla oluşturulan sistemler yetersiz ve eksik bilgilere rağmen doğru sonuçlar çıkarabildiğinden mühendislik uygulamalarında ve birçok alanda tercih edilir. Cep bilgisayarlarında el yazısı algılama teknolojisi (Sony), video kameralarda hareketin algılanması (Canon, Minolta), el yazısı ve ses tanımlama (CSK, Hitachi, Hosai Univ., Ricoh), helikopterler için uçuş desteği (Sugeno), çelik sanayisinde makine hızı ve ısısının kontrolü (Kawasaki Steel, New-Nippon Steel, NKK), dokümanların arşivleme sistemi (Mitsubishi Elec.) bu alanlardan bazılarıdır (Yılmaz, 2006).

3.3. Özniteliklerin Elde Edilmesi

Bu çalışmada dokümanlar, gövde tabanlı, sözcük tabanlı, hece tabanlı ve karakter tabanlı olmak üzere dört farklı kategoride ele alınmıştır. Gövde, sözcük, hece ve karakterler için n-gram analizleri yapılarak öznitelik vektörleri oluşturulmuştur.

Öznitelik vektörlerinin oluşturulması aşamasında kullanılan heceleme algoritması, sözcüklerin kök ve ek analizlerinin nasıl yapıldığı ve n-gram analizi takip eden kısımlarda sırasıyla belirtilmiştir.

3.3.1. Heceleme Algoritması

Doğal Dil İşlemenin temel konularından biri de dillerin işlev ve yapısının daha iyi anlaşılmasıdır. Doğal Dil İşleme ve Veri Madenciliği alanlarında dili oluşturan kelimelerin yapısının, o dili anlamada, karakter analizinin yapılmasında, cümle sınırlarının belirlenmesinde, veri sıkıştırma, kriptoloji, konuşma sentezleme, hece istatistiklerinin oluşturulması gibi alanlarda büyük rolü vardır.

Bir sözcüğün yapısının anlaşılabilmesi için cümlede yer aldığı konuma daha sonra da sözcüğün ek ve köklerine bakılmalıdır. Böylece sözcüğün özne mi yüklem mi

nesne mi olduğu anlaşılır. Sözcük analizinin yapılabilmesi için o sözcüğün ek ve köklerine doğru bir şekilde ayrılması gerekir.

Heceleme algoritmaları; metin karşılaştırma, kelime düzeltme, dil tanıma, heceleme gibi alanlarda kullanılır. Heceleme metotları genelde kök merkezli ve veri tabanlı olmak üzere 2 kategori altında sınıflandırılır (Adsett vd.; Marchand, 2009). Uygulamada kök merkezli yaklaşım kolay olduğundan bazı diller için kök merkezli hecelendirme algoritması tasarlanmıştır. Bazı dillerde ise bütün sözcükleri hecelemek mümkün olmadığı için bu dillerde hece sınırlarının belirlenmesinde veri tabanlı metot kullanılmıştır.

Türkçe, Ural Altay dil ailesinin Altay gurubuna ait 29 harften oluşan bir dildir. Çizelge 3.1.'de Türkçe harflerin yapısı verilmiştir.

Çizelge 3.1. Türkçedeki harflerin yapısı

Sesliler: a e ı i o ö u ü

Sessizler: b c ç d f g ğ h j k l m n p r s ş t v y z

Türkçede heceler en az 1 en fazla 5 harften oluşmaktadır. Türkçedeki hece yapıları Çizelge 3.2.'de verilmiştir. Burada V, sesli harfi; C ise sessiz harfi temsil etmektedir.

Türkçedeki hece yapıları iki kural ile tanımlanır. Kurallardan birincisi; iki sesli harf arasına bir sessiz harf girdiğinde sözcüğün ilk hecesi sözcük ortasından oluşturulur. Bu kuralı tanımlamak için VCV→V-CV yapısı kullanılabilir. Örneğin "araba" sözcüğü "a-ra-ba" olarak hecelenebilir.

İkincisi; bir sözcükte aralarında sesli harf olmaksızın sessizler yan yana geldiğinde, sözcüklerdeki hecelerin başlangıcı, sözcük ortası bir heceyi oluşturacak şekilde devam eder. Örneğin "belge" ve "renkli" sözcüklerini sırası ile heceleyecek olursak "bel-ge" ve "renk-li" olur. Bu kural VC1C2V→VC1-C2V ve VC1C2C3V→VC1C2-C3V olarak gösterilebilir.

Bir sesli harf hece oluşturabilmesine rağmen sessiz harfler sesli harf olmadan hece oluşturamamaktadır. Bundan dolayı sözcüklerdeki sesli harf sayısı hece sayısını verir. Bu durumda sözcükteki sesli harflerin yerlerinin belirlenmesi gerekir.

Türkçede sekiz tane sesli harf bulunduğundan sesli harf dizininin belirlenmesi için her bir karakter üzerinde sekiz tane karşılaştırma işleminin yapılması gerekir.

Çizelge 3.2. Türkçedeki hece yapıları

HECE YAPILARI	ÖRNEK HECELER
V	A, e, ı, i, o, ö, u, ü
VC	Ab, ac, ak, az, ek...
CV	Ba, be, bı, ra...
CVC	Bel, gel, cam, gör, kal, kol...
VCC	Alt, üst, ilk...
CCV	Bre, gri...
CVCC	Kurt, kırk, kalk, renk...
CCVC	Tren, krem...
CCVCC	Tvist...

Türkçe için kullanılan heceleme algoritmaları ile sözcüklerdeki sesli harflerin yeri doğru bir şekilde tespit edilebilir. Literatürde iki farklı algoritma (Ucoluk ve Toroslu, 1997; Akın, 2005) ve çalışmada kullanılan RASAT Heceleme Algoritması (Aşlıyan ve Günel, 2011) vardır. Çizelge 3.3.'de RASAT Heceleme Algoritmasının yapısı verilmiştir.

Çizelge 3.3. RASAT heceleme algoritması

```
// Girdi: alfabe←"λBbCcDdFfGgHhJjKkLlMmNnPpQqRrSsTtVvWwXxYyZz
// ÇçĞğŞsAaEeIıİiÜüUuOoÖö"  /* genel değişken */
1: procedure Construct_HashTable( )
2:        for i = 0 to length of alfabe do
3:                k←alfabe[i] 'yi ASCII koda dönüştür ;
4:                htable[k] = i;
5:        end for
6: end procedure

7: function syllabify(word)
Girdi: s_counter  ← 0  // İlk olarak sıfır. Sözcükteki hece sayısını hesaplar.
υ1← 0, v2 ← 0          // İlk olarak sıfır değer alır. Art arda hecelerin birinci ve
// ikinci seslilerin indisleri.
s_indices[s_counter] ← -1;       // Hece indislerinin listesi.
```

```
and syllable_list ← Boş Liste;
8:                  counter ← 0;    // Bir sözcükteki karakterleri hesaplar

9:          while (counter < sözcük uzunluğu) do
10:                 k ← word[counter] , ASCII koda dönüştür
11:                 if (htable[k] > 48) then
12:                         υ1 ← counter; break;
13:                 end if
14:                 counter ← counter + 1;
15:         end while
16:         first _index ← v1 + 1;    // İlk seslinin indisi bulunur.

17:         for i = ilk indis  to sözcük uzunluğu do
18:                 k ← word[i] ' yi ASCII koda dönüştür
19:                 if (htable[k] < 49) then // Sessiz harf ise

20:                 continue;
21:         else
22:         υ2 ← i;
23:         C_counter   υ2 - v1 - 1; // Hecedeki sessizlerin sayısını hesaplar.

24:         s_counter ← s_counter + 1;
25:         if (C counter = 0) then
26:                 s_ indices[s_counter] ← υ2 - 1;
27:         else
28:                 s_indices[s_counter]   ←υ2 - 2;
29:         end if
30:         substr ← s_indices[s_counter-1]+1 indisinden başlayan ve
s_indices[s_counter] – s_ indices[s_counter - 1] uzunluğundaki karakter
dizisini ata
31:                 substr 'yi  the syllable_list'e ekle
32:                 υ1 ← υ2;
33:                 end if
34:         end for
35:         s_counter ← s_counter + 1;
36:         s_indices[s_counter] ← sözcük uzunluğu -1;
37:         substr ← s_indices[s_counter-1]+1 indisinden başlayan ve
s_indices[s_counter] – s_ indices[s_counter - 1] uzunluğundaki karakter
dizisini ata
38:         substr'yi  syllable list'e ekle
39:         return syllable_list;
40:         end function
```

3.3.2. Morfolojik Analiz

İlk olarak 1859 yılında dilbilimine bir terim olarak giren morfoloji; dilde biçimi oluşturan öğelerin biçimlerini tanımlamak, biçimsel öğeleri sınıflandırmaktır. Türkçe kelimelerin morfolojik analizi ise sözcüklerin ek ve köklerine göre ayrılarak yapısının belirlenmesi işlemidir. Türkçe bir sözcüğü incelediğimizde kök ve eklerden oluştuğunu görürüz. Türkçe sondan eklemeli bir dil olduğundan yeni sözcükler, sözcük kök ve gövdelerine ekler eklenmesi ile oluşur. Bir sözcükte anlamlı en küçük parça kök ve bundan sonra gelenler ise ektir. Sözcüklerin morfolojik analizi; Kök Analizi ve Ek Analizi olmak üzere iki kısımda yapılır.

Kök Analizi: Türkçe bir sözcüğün kökünün bulunabilmesi için öncelikle kök veri tabanının oluşturulması gerekir. Daha sonra bir sözcüğün kökünü bulabilmek için sağdan veya soldan başlayarak kök veritabanındaki köklerle girilen sözcük karşılaştırılır. Fakat bulunan kök sözcüğün gerçek kökü olmayabilir. Örneğin bu sözcük ‘balıkçılık’ olsun. Sözcüğün gerçek kökü ‘balık’ olmasına rağmen veri tabanında balık kökü olmasın veya ‘bal’ kökü ‘balık’ kökünden önce gelsin. Bu durumda program sözcüğün kökünü ‘bal’ olarak kabul eder. Bu problemin ortadan kalkabilmesi için ek analizinin yapılması gerekir.

Ek Analizi: Türkçe sözcüklerde her bir kökün türüne göre sözcüğe eklenecek olan ekler ve eklerin de çeşitlerine ve yapılarına göre ek gurupları oluşturulur. Böylece kök analizi yapılırken ek analizi de yapılarak gerçek kök bulunur. Türkçe sözcüklerde ekler kökün türüne göre sözcüklere eklenir.

Türkçe sözcüklerde kök ve gövdelerin belirlenmesinde bazı metotlar kullanılmıştır. Bunlar:

1. AF Algoritması (Solak ve Can, 1994)
2. Longest-Match (L-M) Algoritması (Kut vd, 1995)
3. Identified Maximum Match Algoritması (Köksal, 1975)
4. FindStem Algoritması
5. Solak and Oflazer Algoritması (Solak ve Oflazer, 1993)

6. Root Reaching Method without Dictionary (Cebiroğlu ve Adalı, 2002)

7. Extended Finite State Approach (Oflazer, 1999)

Çalışmada sözcüklerin olası kök ve ekleri belirlenirken Longest-Match (L-M) Algoritması kullanılmıştır.

3.3.3. N-Gram Analizi

Yapay Zekânın alt dallarından biri olan Doğal Dil İşleme (Natural Language Processing) (NLP) çalışmalarının temel amaçlarından biri dillerin işlev ve yapısının daha iyi anlaşılmasıdır. NLP bilgi şifreleme, konuşma tanıma (Marchand vd. 2009), optik karakter belirleme, yazı doğrulama gibi pek çok alanda kullanılır. Bu alandaki ilk çalışmalar Shannon ve Zipf tarafından 1940'lı yıllarda başlamış olup birçok araştırmacı tarafından geliştirilmiştir. Yapılan çalışmalarla bir sözcüğe göre ardından gelecek olan başka bir sözcük tahmin edilerek sözcüklerin n-gram analizleri yapılmıştır.

Sözcükler tek başlarına anlam içerebildikleri gibi yanlarına gelen farklı sözcüklerle farklı anlamlar da içerebilirler. Böyle bir durumda sözcüklerin n-gram analizlerinin yapılması anlam belirsizliğini ortadan kaldırmada yardımcı olur. n-gram analizi bir dizideki tekrar oranını bulmaya yarayan bir yöntemdir. Buradaki 'n' tekrar sayısı olmak üzere genelde literatürde 1-gram(unigram), 2-gram(bigram) ve 3-gram(trigram) analizleri kullanılır. n-gram analizi ile cümle içerisinde sözcük, sözcük içerinde de hece olasılıkları hesaplanır.

Verilen bir metindeki sözcükler kümesi, $J = \{S_1, S_2, \dots, S_j, \dots, S_n\}$ kümesi olarak kabul edilebilir.

S_j herhangi bir sözcüğü temsil etmektedir ve n metindeki sözcük sayısını ifade etmektedir. S_j sözcüğünün, $H = \{h_1, h_2, \dots, h_i, \dots, h_t\}$ hece dizisinden oluştuğunu varsayalım ve bu hece dizisinin 2-gramlarını hesaplayalım. $1 \leq i \leq t$ için S_j sözcüğündeki h_i hecesinin olasılığının kendisinden önce gelen h_{i-1} hecesine bağlı olduğu varsayılır. Bu durumda S_i sözcüğündeki h_i hecesinin h_{i-1} den sonra gelme olasılığı $P(h_i|h_{i-1})$ Denklem 3.11'deki gibi hesaplanır.

$$P(h_i|h_{i-1}) = \frac{C(h_{i-1},h_i)}{C(h_{i-1})} \tag{3.11}$$

Denklem 3.11'de verilen $C(h_{i-1})$ değeri h_{i-1} hecesinin, $C(h_{i-1}, h_i)$ değeri ise $h_{i-1}h_i$ hecesinin toplam frekansını vermektedir.

Öznitelik uzayının oluşturulmasından sonra sınıflandırma metotlarından biri kullanılabilir. Bu çalışma da sınıflandırma metodu olarak K-En Yakın Komşu Algoritması, MLP ve SVM ağları kullanılmıştır.

3.4. Kullanılan Yöntemler

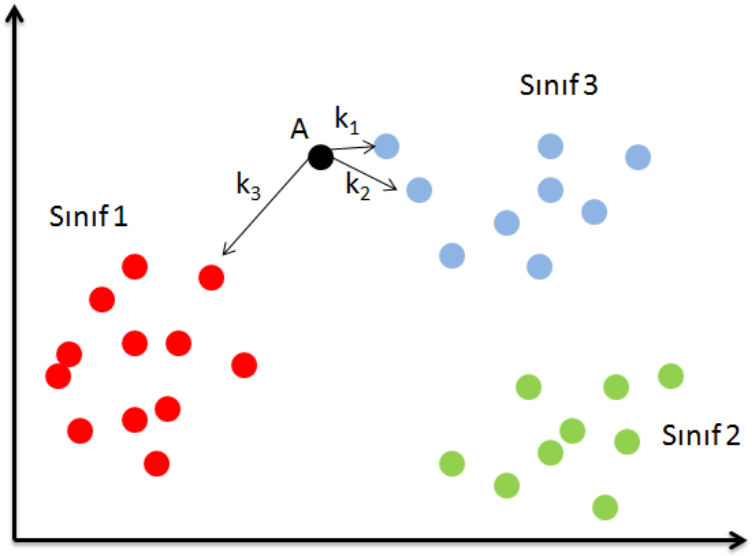
3.4.1. K-En Yakın Komşu Modeli (K-NN)

Bellek tabanlı sınıflandırma yöntemi olan K-En Yakın Komşu modelinde (K-NN) bir elemanın sınıfını belirlemek için önceden sınıfları belirlenmiş olan eğitim kümesindeki k tane elemandan yararlanılır. Sınıfı belirlenmek istenen bir eleman her bir sınıftaki elemanla karşılaştırılır. Bu elemana en yakın k tanesi seçilir. Seçilen elemanlar en çok hangi sınıfa ait ise sınıflandırmak istediğimiz eleman da o sınıfa aittir. Uzaklığın hesaplanmasında Öklid uzaklık formülü kullanılır. x ve y noktaları için Öklid uzaklık formülü (Özkan, 2008) Denklem 3.12' verilmiştir.

$$d(x,y) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2} \qquad (3.12)$$

K-NN algoritması oluşturulurken, ilk olarak k değeri belirlenir. Bu değer sınıfı belirlenmek istenen örneğin en yakın komşularının sayısıdır. Ardından örneğin eğitim kümesindeki bütün noktalara olan uzaklıkları ayrı ayrı hesaplanır. Bu uzaklıklardan en küçük olan k tanesi seçilir. Seçilen k tane eleman en çok hangi sınıfa ait ise örnek de o sınıfa alınır. K-NN modelinde sınıflandırılmak istenen örneğin eğitim kümesindeki tüm elemanlarla uzaklığı hesaplandığından sınıflandırma işlemi uzun sürer fakat eğitim kısa sürede tamamlanır.

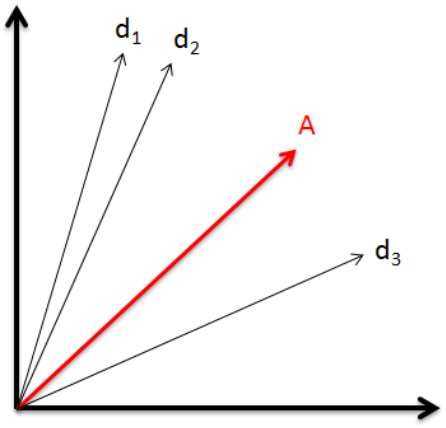
$k = 3$ için bir A örneği K-NN'ye göre sınıfının belirlenmesi Şekil 3.2.'de gösterilecek olursa;



Şekil 3.2. "A" örneğinin K-NN'ye göre sınıfının belirlenmesi

3.4.2. K-NN Algoritmasının Doküman Sınıflandırmada Uygulanması

Çalışmada kullanılacak olan sınıflar, dokümanlar ve bu dokümanlardaki sözcükler vektör haline dönüştürülerek yukarıda bahsettiğimiz adımlar gerçekleştirilir. Dokümanların vektör uzayı modelinde gösterimi Şekil 3.3.'de verilmektedir (Pilavcılar, 2007).



Şekil 3.3. Dokümanların vektör uzayı modeli

Örnek olarak “Vücutta vazgeçilmez bir madde olan vitaminlerin bir kısmı vücutta üretilirken çoğu vücutta üretilemez.” metnini ele alalım. Bu metnin vektör tablosu Çizelge 3.4.' de gösterildiği gibidir.

Çizelge 3.4. Metnin vektör tablosu

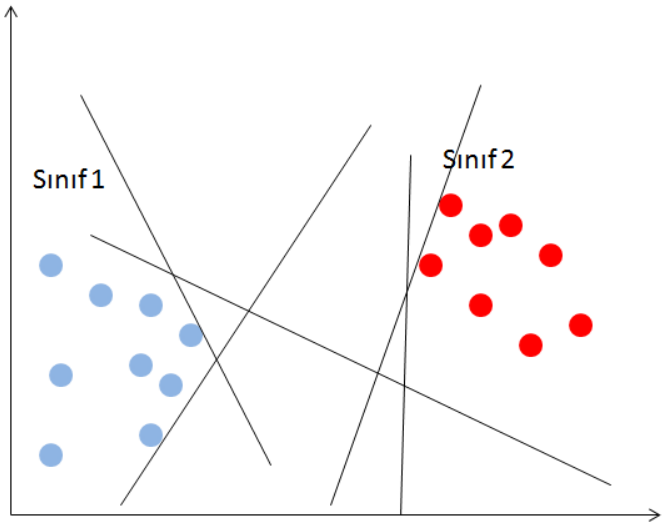
Sözcükler	Vücut	Vazgeç	Bir	Madde	Olan	Vitamin	Kısmı	Üret	Çoğu
Frekanslar	3	1	2	1	1	1	1	2	1

9 boyutlu olan bu metin (3,1,2,1,1,1,1,2,1) vektörü olarak gösterilir.

Metindeki ağırlıkları hesaplanan sözcükler ile eğitim kümesindeki daha önceden belirlenmiş olan sözcüklerinki karşılaştırılarak uzaklıklar hesaplanır.

3.4.3 Destek Vektör Makinesi (SVM)

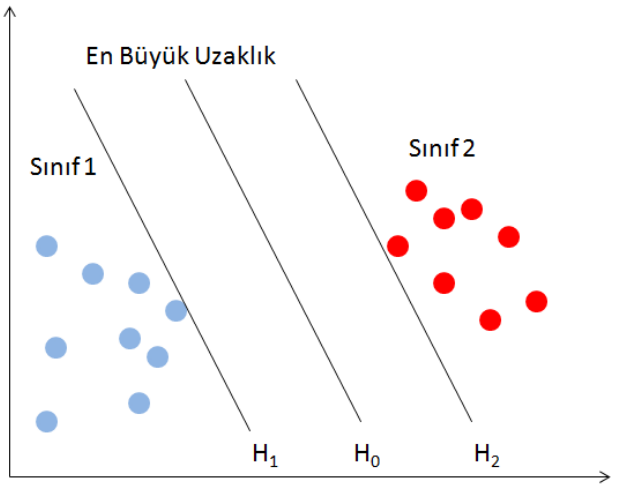
Optimizasyona dayalı bir sınıflandırma modelidir. Destek vektör makinesi sınıflandırma yaparken doğrusal ya da doğrusal olmayan verileri en uygun karar fonksiyonunun yardımıyla gerçekleştirir. En basitinden iki sınıf ele alalım. İki sınıfı birbirinden ayıran birçok hiper düzlem vardır. Şekil 3.4.'de iki sınıf için doğrusal ayrılabilen verilerin hiper düzlemleri verilmiştir. Destek vektör makinesinin amacı bu hiper düzlemlerden iki sınıfı en iyi ayıran hiper düzlemi belirlemektir. Bunu da karar fonksiyonunun yardımıyla gerçekleştirir. Destek vektör makinesi verilerin doğrusal olarak ayrılabilme ve doğrusal olarak ayrılamama durumlarına göre iki farklı şekilde sınıflandırma yapar.



Şekil 3.4. İki sınıf için doğrusal ayrılabilen verilerin hiper düzlemleri

3.4.4. Doğrusal Olarak Ayrılabilen Veriler İçin Sınıflandırma

İki boyutlu uzayda doğrusal veriler için iki sınıf göz önüne alalım. $H = \{H_1, H_2, \ldots H_N\}$ hiper düzlemler olmak üzere iki sınıf için kendisine en yakın noktalar arasındaki en büyük uzaklığı hesaplanan iki hiper düzlemi belirler. Şekil 3.5'de görülen H_0 düzlemi iki sınıfı birbirinden ayıran doğrusal hiper düzlemdir.



Şekil 3.5. Doğrusal ayrılabilen verilerin arasındaki en büyük uzaklık

$y \in \{-1, +1\}$ ile etiketlenen iki sınıfın $i = 1,2, \dots n$ olmak üzere n tane örnek için $\{x_i, y_i\}$ değerleri veri kümesi olsun. İki sınıfı ayıran ve birbirine paralel H_1 ve H_2 hiper düzlemleri Denklem 3.13 ve Denklem 3.14'deki gibi ifade edilmiştir:

$$H_1 = W^T X + b = 1 \tag{3.13}$$

$$H_2 = W^T X + b = -1 \tag{3.14}$$

Burada $W = \{w_1, w_2, \dots w_n\}$ ağırlık vektörünü, iki boyutlu uzayı ele aldığımız için $X = \{x_1, x_2\}$'yi ve b değeri de sabit sayıyı göstermektedir. Bu durumda H_1 ve H_2 düzlemlerine paralel ve bu düzlemlere eşit uzaklıkta olan H_0 düzlemi Denklem 3.15'teki gibi ifade edilir:

$$H_0 = W^T X + b = 0 \tag{3.15}$$

H_0 hiper düzleminin üst tarafında ve alt tarafında kalan noktalar sırasıyla Denklem 3.16 ve Denklem 3.17'deki eşitsizliğe uyar.

$$W^T X + b > 0, y_1 = +1 \tag{3.16}$$

$$W^T X + b < 0, y_1 = -1 \tag{3.17}$$

Sonuç olarak $\forall_i$ değeri için Denklem 3.18'deki gibi sonuç elde edilir.

$$y_i(W^T X + b) - 1 \geq 0 \tag{3.18}$$

Burada destek vektör makinesinin amacı H_1 ve H_2 hiper düzlemleri arasındaki uzaklığı maksimuma çıkarmaktır. Bunun için de H_1 veya H_2 hiper düzlemlerinin H_0 hiper düzlemi üzerindeki bir P noktasına olan uzaklığı Denklem 3.19'daki gibi hesaplanır.

$$d = \frac{|WX_p' \mp b|}{\|W\|} = \frac{|w_1 x_{1p} + w_2 x_{2p} + \cdots + w_n x_{np} + b|}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} \tag{3.19}$$

3.4.5 Çok Katmanlı Algılayıcı Modeli (MLP)

Yapay sinir ağları insan beyninin çalışma prensibini örnek alarak oluşturulmuş yapay öğrenme metotlarıdır. İnsan beyninin öğrenme yolu ile yeni bilgiler

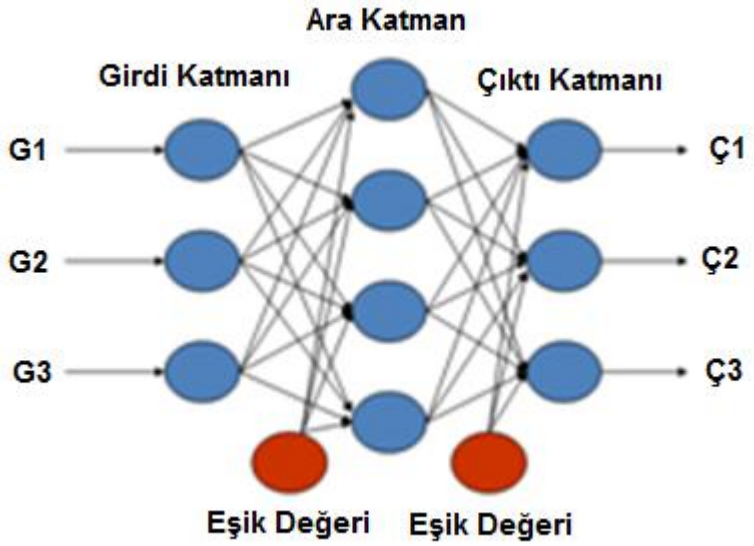
üretebilme, keşfedebilme, mevcut bilgiler ile olaylar hakkında yorum yapabilme, karar verebilme, olaylar arasında ilişki kurabilme gibi özelliklerini yapabilmek için tasarlanmıştır. Genellikle doğrusal olmayan, çok boyutlu, karmaşık, eksik, hatalı verilerin olması durumunda tercih edilirler. Günümüzde tıpta, finansal konularda, mühendislik problemlerinin çözümünde, veri madenciliğinde, optik karakter tanımada, güvenlik sistemlerinde konuşma ve parmak izi tanımada, radar ve sonar sistemleri sınıflandırmada, kan analizlerinin sınıflandırılmasında, kanser tespitinde, kalp krizi tedavisinde, beyin modellemesi çalışmalarında yaygın olarak kullanılırlar.

Yapay sinir ağları insanlarda olduğu gibi örnekler sayesinde öğrenirler ve daha önce hiç görmediği olaylar hakkında yorum yapabilirler. İnsanlardaki sinir hücrelerinin yapay sinir ağlarındaki karşılığı proses elemanlarıdır. İnsanlarda sinir sistemi birçok sinir hücresinin birleşmesiyle oluşur ve öğrenme bu sinir hücreleri arasındaki sinaptik boşluklarda elektriksel ayarlamalarla gerçekleşir. Yapay sinir ağlarında ise ağa verilen girdilerin ağırlıklarının ve buna bağlı olarak ağın yapısını değiştirmesiyle gerçekleşir. Yapay sinir ağları öğretmenli öğrenme, öğretmensiz öğrenme ve destekleyici öğrenme olmak üzere üç farklı öğrenme tipine sahiptir. Çalışma gerçekleştirilirken öğretmenli öğrenme metotlarından çok katmanlı algılayıcı metodu kullanılmıştır.

3.4.6. Çok Katmanlı Algılayıcı İle Sınıflandırma

Yapay sinir ağlarının ilk modelleri aralarında doğrusal olan olayları çözebilmekteydi. Girdi ve çıktıları arasında doğrusal ilişki bulunmayan olayların çözümünde bunlar yeterli olmayıp çok katmanlı algılayıcı modeli geliştirilmiştir. Çok katmanlı algılayıcı (MLP) modelinin amacı ağın çıktısı ile beklenen çıktı arasındaki hatayı en aza indirmektir. Bunu gerçekleştirirken de hatayı ağa yayar. Bundan dolayı bu ağa hata yayma ağı da denmektedir. Çok katmanlı algılayıcı ağı öğretmenli öğrenme metoduna göre öğrenir. Ağa hem girdiler hem de bu girdilerden elde etmesi gereken çıktılar verilir. Çok katmanlı algılayıcı ağı 1girdi katmanı, 1 veya daha fazla ara katman, 1 de çıktı katmanından oluşur. Şekil 3.6.'de basit bir MLP modelinin yapısı verilmiştir (Öztemel, 2003). Dışarıdan sisteme verilen bilgiler hiçbir işleme tabi tutulmadan ara katmana iletildiği için buradaki k tane proses elemanı için girdi katmanının çıktısı $Ç_k^i$ Denklem 3.20'de görüldüğü gibi olur.

$$Ç_k^i = G_k \tag{3.20}$$



Şekil 3.6. MLP modelinin yapısı

Ara katmanın çıktısı Denklem 3.21'de gösterildiği üzere girdi katmanındaki her bir proses elemanından gelen çıktı ile bunların $A = \{A_1, A_2, \dots\}$ ağırlıklarının çarpımının toplanması sonucu elde edilir.

$$NET_j^a = \sum_{k=1}^{n} A_{kj} Ç_k^j \tag{3.21}$$

Denklem 3.21'de verilen A_{kj} değeri girdi katmanındaki k. proses elemanını ara katmandaki j. proses elemanına bağlayan bağıntının ağırlığıdır. Ara katmandaki j. proses elemanının çıktısı ise bu katmana gelen NET girdinin aktivasyon fonksiyonundan geçirilmesi ile elde edilir. Aktivasyon fonksiyonu olarak Çizelge 3.5.'de verilen fonksiyonlar kullanılabilir. Çalışmada aktivasyon fonksiyonlarından sigmoid fonksiyonu kullanılmıştır. Sonuç olarak ara katmandaki j. proses elemanının çıktısı Denklem 3.22'daki gibi olur.

$$Ç_j^a = \frac{1}{1+e^{-(NET_j^a + B_j^a)}} \tag{3.22}$$

Çizelge 3.5. Aktivasyon fonksiyonları

Lineer fonksiyon	$F(NET) = NET$
Step fonksiyonu	$F(NET) = \begin{cases} 1, & NET > eşik\ değer \\ 0, & NET \leq eşik\ değer \end{cases}$
Sinüs fonksiyonu	$F(NET) = sin(NET)$
Eşik değer fonksiyonu	$f(x) = \begin{cases} 0, & NET \leq 0 \\ NET, & 0 < NET < 1 \\ 1, & NET \geq 1 \end{cases}$
Hiperbolik tanjant fonksiyonu	$F(NET) = (e^{NET} + e^{-NET})/(e^{NET} - e^{-NET})$

Aktivasyon fonksiyonundaki B_j değeri ara katmandaki j. elemana bağlanan eşik değerin ağırlığıdır. Elde edilen çıktı ile beklenen çıktı arasındaki fark, hatayı verir. Bu hata geriye doğru ağa yayılarak minimuma düşünceye kadar ağın ağırlıkları değiştirilir.

$(B_1, B_2, \dots)$ ağın beklenen çıktıları, $(Ç_1, Ç_2, \dots)$ ağın çıktısı olmak üzere çıktı katmanındaki m. proses elemanında oluşan hata Denklem 3.23'de verilmiştir.

$$E_m = B_m - Ç_m \tag{3.23}$$

4. SİSTEMİN UYGULANMASI VE BULGULAR

4.1. Veri Tabanının Oluşturulması

Sistemin uygulanmasında ilk olarak gerçekleştirilen adım, veri tabanının oluşturulmasıdır. Veri tabanının oluşturulmasında öznitelik vektör uzayında yer alacak olan sözcükler belirlenmiştir. Bu sözcükler için her bir sınıfa ait 75 doküman birleştirilmiştir. Eşik değeri olarak da 0,25, 0,50, 0,75, 0,90 değerleri alınarak dört farklı yöntem ile veri tabanı oluşturulmuştur. Bu yöntemlerden biri olan Longest-Match (en uzun gövdeleme) algoritması ile dokümanlardaki sözcükler gövdelere ayrılmış ve gövde 1-gram, 2-gram ve 3-gramlarına göre veri tabanı oluşturulmuştur. Kullanılan başka bir yöntem de heceleme algoritmasıdır. Heceleme algoritmasına göre dokümanlardaki sözcükler hecelere ayrılmış ve hece 1-gram, 2-gram ve 3-gramlarına göre veri tabanı oluşturulmuştur. Diğer iki yöntem ise karakter analizi ve sözcük analizidir. Karakter analizinde dokümanlardaki sözcükler karakterlere ayrılmış ve karakter 2-gram, 3-gram, 4-gram, 5-gram ve 6-gramları oluşturularak veri tabanı elde edilmiştir. Sözcük analizinde ise dokümanlar sözcük bazında ele alınmış olup sözcük 1-gram, 2-gram ve 3-gramlarına göre oluşturulmuştur.

4.2. Sistemin Değerlendirilmesi

Bu çalışmada, sonuçların değerlendirilmesinde Kesinlik skoru (Precision), Hassasiyet skoru (Recall), ortalama doğruluk (Accuracy) değerleri ve F-ölçüsü (F-measure) değerleri kullanılmıştır.

Doküman sınıflandırmada çok sık karşılaşılan kıyaslama yöntemlerinden olan kesinlik skoru ve hassasiyet skoru değerlerinin hesaplanmasında kullanılan Doğru Pozitif (DP), Doğru Negatif (DN), Yanlış Pozitif (YP) ve Yanlış Negatif (FN) değerleri sonucu "Pozitif" ve "Negatif" olan iki sınıf için dört farklı olası sonuçları verir. DP, pozitif sınıfında olan dokümanların, pozitif olarak sınıflandırılmış dokümanların sayısıdır, yani doğru sınıflandırmış olur. YP, negatif sınıfında olan dokümanların, pozitif olarak sınıflandırılmış dokümanların sayısıdır, yani yanlış sınıflandırmış olur. YN, pozitif sınıfında olan dokümanların, negatif olarak sınıflandırılmış dokümanların sayısıdır; yanlış sınıflandırılır. DN, negatif sınıfında

olan dokümanların, negatif olarak sınıflandırılmış dokümanların sayısıdır; doğru sınıflandırılmış olur.

DP, pozitif sınıfındaki dokümanların doğru sınıflandırma sayısını, DN ise negatif sınıfındaki dokümanların doğru sınıflandırma sayısını gösterir. Her iki durumda da doğru sınıflandırma yapmış olunur. Bu olasılık sonuçları Çizelge 4.1.'de hata matrisi olarak adlandırılan bir matrisle gösterilir. Hata matrisinde köşegen elemanları doğru tanıma sayısını, köşegen dışı elemanlar ise yanlış karar sayısını gösterir.

Çizelge 4.1. Hata matrisi

	"Pozitif" sınıfı	**"Negatif" Sınıfı**
Test Sonucu "Pozitif"	DP	YP
Test Sonucu "Negatif"	YN	DN

Kesinlik skoru bir sınıftaki doğru olarak sınıflandırılan dokümanların sayısının, o sınıftaki toplam doküman sayısına oranını; hassasiyet skoru ise bir sınıftaki doğru olarak sınıflandırılan dokümanların sayısının, sistemin o sınıf olarak tespit ettiği toplam doküman sayısına oranını verir.

Kesinlik skoru ve hassasiyet skoru değerlerini Denklem 4.1 ve Denklem 4.2'de gösterecek olursak:

$$Kesinlik\ skoru = \frac{DP}{DP+YP} \tag{4.1}$$

$$Hassasiyet\ skoru = \frac{DP}{DP+YN} \tag{4.2}$$

Kesinlik skoru ve hassasiyet skoru değerlerinin her ikisinin de aynı anda iyi olması amacıyla F-ölçüsü değeri hesaplanır. F-ölçüsü değeri Denklem 4.3'deki gibi hesaplanır.

$$F - \text{ölçüsü} = \frac{2 \times Kesinlik\ skoru \times Hassasiyet\ skoru}{Kesinlik\ skoru + Hassasiyet\ skoru} \quad (4.3)$$

4.3. Sistemin Eğitilmesi ve Test Edilmesi

Sistemin eğitilmesinde her bir sınıfa ait 75 dokümandan 25 tanesi toplamda 150 doküman, test aşamasında ise daha önce sistemin görmediği diğer 50 doküman, toplamda 300 doküman sistemi verilmiştir. Eğitim ve test aşamalarında K-En Yakın Komşu modeli, Çok Katmanlı Algılayıcı ağı ve Destek Vektör Makinesi olmak üzere üç farklı yöntem kullanılmıştır. En Yakın komşu modelinde en yakın k değerleri olarak 1, 3, 5 ve 7 değerleri alınmıştır. Çizelge 4.1.'de K-NN'ye göre ortalama doğruluk değerleri Çizelge 4.2.'de K-NN'ye göre ortalama F-ölçüsü değerleri verilmiştir.

Çizelge 4.2. K-NN'ye göre ortalama doğruluk değerleri

		Eşik değeri	**0,25**	**0,50**	**0,75**	**0,90**
		k-değerleri				
Gövde tabanlı	**1-gram**	**1**	90,3	89,8	87,9	87,3
		3	90,6	89,3	87,9	87,6
		5	**90,4**	88	87,1	86,8
		7	89,9	87,2	86,9	86,6
	2-gram	**1**	77,8	76,6	75,8	76
		3	77,3	78,1	77,2	76,4
		5	79,1	77,1	74,7	74,4
		7	78,6	76,7	74,2	73,8
	3-gram	**1**	76,8	76,3	75,7	75
		3	74,1	73,9	73,3	73,2
		5	74,7	73,7	73	72,9
		7	74,2	74,2,	72,9	72,8
Sözcük tabanlı	**1-gram**	**1**	**87,1**	83,4	84,3	84
		3	86,7	83,4	83,8	83,4
		5	85,1	82,8	83,2	83,2
		7	84	82,4	82,8	82,9
	2-gram	**1**	75,1	76,3	75,3	75,4
		3	74,2	76,8	74,6	74,9
		5	75,8	76,2	74,2	73,3
		7	76,8	76,7	73,9	73,1
	3-gram	**1**	76,2	75,9	75,7	75,6
		3	74	73,9	73,7	74,3

Çizelge 4.2. K-NN'ye göre ortalama doğruluk değerleri (Devamı)

		5	73,2	73,8	73,8	72,4
		7	73,2	72,2	72,2	71,1
Hece tabanlı	**1-gram**	**1**	93,3	92,3	89,3	88,1
		3	94,4	92,4	88,9	88,1
		5	94,9	92,4	89,8	86,9
		7	**95,4**	93	89,2	87,6
	2-gram	**1**	89,9	89,1	86,6	86,3
		3	90	87,9	86	84,7
		5	90	87	84,9	83,9
		7	89,1	86,3	84,2	83,8
	3-gram	**1**	87,1	84,6	81,6	81
		3	85,3	82	79,3	78,6
		5	84,8	80,7	78,6	78,2
		7	84,3	82,2	80,6	80
Karakter tabanlı	**2-gram**	**1**	89,6	90,6	88,7	89,6
		3	90,7	90,9	89,1	88,1
		5	90,8	91,4	88,7	87,9
		7	90,6	91,3	88,7	87
	3-gram	**1**	94,9	94,2	90,3	88,9
		3	**95,6**	95	89,7	87,8
		5	94,7	94	87,7	87,2
		7	94,6	93,9	88,7	87
	4-gram	**1**	92,3	92,3	92,9	92
		3	91,9	91,3	91,8	91,7
		5	91,3	89,8	91,6	91,3
		7	89,3	88,4	91,2	91,1
	5-gram	**1**	90	88,9	87,8	88
		3	88,7	87,2	86,6	87,2
		5	86,7	84,9	85,6	85,8
		7	84,3	82,7	83,8	84,8
	6-gram	**1**	84	79,4	77,8	76,6
		3	80,2	76	74,7	74,2
		5	76,8	75,1	74,3	74,1
		7	75,4	74,7	74,1	73,9

Çizelge 4.3. K-NN'ye göre ortalama F-ölçüsü değerleri

		Eşik değeri	**0,25**	**0,50**	**0,75**	**0,90**
		k-değerleri				
Gövde tabanlı	**1-gram**	**1**	68	65,4	58,6	55,8
		3	**68,7**	63,6	56,9	54,8
		5	67,4	57,9	52,7	51,1
		7	66	55,3	51,7	50,7
	2-gram	**1**	32,5	26,7	23,1	24
		3	30,6	32,6	28,7	25,7
		5	34,7	25,8	16,2	16
		7	31,2	22,6	14,6	12,7
	3-gram	**1**	27,8	26,4	23,6	20,4
		3	15,3	14,7	11,7	11
		5	18,2	13	9,2	8,6
		7	16	16,3	8,5	8
Sözcük tabanlı	**1-gram**	**1**	**55,9**	45,8	44,6	42,2
		3	52,9	44,4	38,9	40,1
		5	46,9	40,2	38,1	40,1
		7	42,5	39,4	37,6	39,2
	2-gram	**1**	20,3	25,2	20,8	21,8
		3	15,9	27	17,5	19,1
		5	22,7	24,8	15,3	10,7
		7	24,7	26	13,5	9,5
	3-gram	**1**	25,1	23,9	22,9	22,2
		3	14,8	14,1	12,9	23,6
		5	10,6	17,9	17,6	13,9
		7	14,1	12,9	12,8	7,1
Hece tabanlı	**1-gram**	**1**	79,4	76,4	64,8	61
		3	83,2	76,4	63,4	61
		5	84,5	76,5	65,5	54,8
		7	**86,1**	78,4	63,3	56,1
	2-gram	**1**	66,6	63,9	55	53,6
		3	67,1	60,9	53,1	48,1
		5	67,1	55,1	49,2	45,5
		7	63,2	52,7	47,1	44,8
	3-gram	**1**	56,3	51,3	42,6	40,4
		3	50,9	42,7	34,3	31,1
		5	49,1	37,7	31,4	29,4
		7	46,3	40,4	35,1	34
Karakter tabanlı	**2-gram**	**1**	68,9	72,3	62,2	64,1
		3	72,1	73,3	63,1	60
		5	72,4	74,7	62,2	58

Çizelge 4.3. K-NN'ye göre ortalama F-ölçüsü değerleri (Devamı)

		7	71,8	74,6	62,3	55,8
	3-gram	**1**	85,1	82,3	66,1	60,9
		3	**86,9**	85,1	64,9	56,7
		5	84,4	81,9	61,1	54,7
		7	84,5	81,5	60,6	54,3
	4-gram	**1**	76,6	74,5	74,8	70,9
		3	76	71,1	70,5	69,5
		5	75,1	66,2	68,8	68,7
		7	69,3	62,4	67,2	67,4
	5-gram	**1**	69,8	64,6	58,4	58,9
		3	67,4	61,4	54,6	55,4
		5	62,1	55	51,8	49,2
		7	55	48,2	45,3	44,3
	6-gram	**1**	50,2	37	31,4	26,3
		3	40,4	23,9	17,7	15,6
		5	28	19,9	16,2	15,1
		7	21,8	17,8	15	13,4

Bu çalışmada gerçekleştirilen yapay sinir ağı 1 girdi katmanı, 3 ara katman ve bir de çıktı katmanından oluşmaktadır. Çok Katmanlı Algılayıcı modelinin sisteme uygulanması ile 200000 iterasyon sonucu sistemin eğitimi tamamlanmış ve test edilmiştir. Çizelge 4.3.'te MLP'ye göre ortalama doğruluk değerleri Çizelge 4.4.'de ise MLP'ye göre ortalama F-ölçüsü değerleri verilmiştir.

Çizelge 4.4. MLP' ye göre ortalama doğruluk değerleri

Eşik değer		**0,25**	**0,50**	**0,75**	**0,90**
En Uzun Gövde	**1-gram**	**96,9**	95,7	94,7	96,4
	2-gram	87,9	84,2	86,9	87,7
	3-gram	80,9	78,7	79,1	77,6
Sözcük n-gramları	**1-gram**	**97**	96,3	94,7	94,7
	2-gram	88	84,3	88,7	84,7
	3-gram	81,6	79,8	78,1	75,9

Çizelge 4.4. MLP' ye göre ortalama doğruluk değerleri (Devamı)

Hece n-gramları	**1-gram**	98,2	98,4	97	96,6
	2-gram	**98,4**	96,8	96,7	97,6
	3-gram	94,2	91,3	91,6	90,8
Karakter n-gramları	**2-gram**	92	93,1	94,1	95,1
	3-gram	98,2	95,6	97,1	96,3
	4-gram	**99**	97,4	96,8	95,8
	5-gram	86,9	91,4	88,4	91,6
	6-gram	78,3	74,7	76,8	74,7

Çizelge 4.5. MLP'ye göre ortalama F-ölçüsü değerleri

Eşik değer		**0,25**	**0,50**	**0,75**	**0,90**
En Uzun Gövde	**1-gram**	**90,7**	87,1	83,3	89,3
	2-gram	64,5	52,4	62	62,7
	3-gram	38,6	28,8	38,7	28,6
Sözcük n-gramları	**1-gram**	**91**	88,9	83,9	83,9
	2-gram	64,4	49	66,6	54,5
	3-gram	45,5	37,5	34,7	27,2
Hece n-gramları	**1-gram**	94,6	**95,3**	91	89,6
	2-gram	85,3	90,4	90	92,6
	3-gram	82,7	73	73,4	71,1
Karakter n-gramları	**2-gram**	75,9	78,7	82,4	85,3
	3-gram	94,7	86,3	91,3	89
	4-gram	**97**	92,1	90,1	86,9
	5-gram	53,5	70	58,5	70,1
	6-gram	28,4	20,3	25,3	18,2

Destek vektör makinesinin sisteme uygulanması ile elde edilen ortalama doğruluk değeri sonuçları Çizelge 4.5.'de, ortalama F-ölçüsü değerleri ise Çizelge 4.6.'da verilmiştir.

Çizelge 4.6 SVM'e göre ortalama doğruluk değerleri

Eşik Değer		**0,25**	**0,50**	**0,75**	**0,90**
En Uzun Gövde	**1-gram**	**99,8**	99,6	99,3	99,2
	2-gram	99,2	99	99	98,7
	3-gram	72,2	72,2	72,2	72,2
Sözcük n-gramları	**1-gram**	99,6	**99,9**	99,2	99,4
	2-gram	98,9	98,4	98	97,7
	3-gram	92,7	72,2	72,2	72,2
Hece n-gramları	**1-gram**	97,8	98,7	98,3	98,6
	2-gram	**99,8**	98,9	99,2	99,3
	3-gram	99,7	99,3	99	99,2
Karakter n-gramları	**2-gram**	-	92,9	72,3	95,2
	3-gram	95,2	98,3	98,8	99,3
	4-gram	**99,8**	99,1	99,3	99,7
	5-gram	99,7	99,7	99	99,3
	6-gram	98,7	98,4	98,2	98

Çizelge 4.7. SVM'ye göre ortalama F-ölçüsü değerleri

Eşik Değer		**0,25**	**0,50**	**0,75**	**0,90**
En Uzun Gövde	**1-gram**	**99,3**	98,7	98	97,7
	2-gram	97,6	97	97	96
	3-gram	4,8	4,8	4,8	4,8
Sözcük n-gramları	**1-gram**	98,7	**99,7**	97,7	98,3
	2-gram	96,7	95,4	93,9	92,9
	3-gram	77,8	4,8	4,8	4,8
Hece n-gramları	**1-gram**	93,2	96	94,9	95,8
	2-gram	**99,3**	96,7	97,7	98
	3-gram	99	98	97	97,7
Karakter n-gramları	**2-gram**	-	77,9	5,4	85,5
	3-gram	85,5	95	96,3	98
	4-gram	**99,3**	97,3	98	99
	5-gram	99	99	97	98
	6-gram	96	95,4	94,7	94,1

Bu çalışmada, K-NN, MLP, SVM metotları kullanılarak dokümanlar gövde n-gramları, sözcük n-gramları, hece n-gramları ve karakter n-gramları olmak üzere 4 farklı kategoriye göre sınıflandırılmıştır. Öznitelik vektörlerinin oluşturulmasında eşik değer olarak 0,25, 0,50, 0,75 ve 0,90 değerleri alınmıştır. K-NN metodunun sisteme uygulanmasında en yakın k değeri olarak 1, 3, 5 ve 7 değerleri alınarak

sonuçlar karşılaştırılmıştır. MLP metodu uygulandığında 200000 iterasyon sonucu ve ayrıca SVM metodu uygulanarak sonuçlar elde edilmiştir.

K-NN metodu kullanılarak yapılan çalışmalarda Çizelge 4.2 ve Çizelge 4.3'den elde edilen sonuçlara göre gövdelerde en yüksek değerler gövde 1-gramlarda k değeri 3, eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 90,4, ortalama F-ölçüsü değeri 68,7 olmaktadır. Sözcüklerde en yüksek değerler sözcük 1-gramlarda k değeri 1, eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 87,1, ortalama F-ölçüsü değeri 55,9 olarak elde edilmiştir. Hecelerde en yüksek değerler hece 1-gramlarda k değeri 7, eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 95,4, ortalama F-ölçüsü değeri 86,1 ve karakterlerde en yüksek değerler ise karakter 3-gramlarda k değeri 3, eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 95,6, ortalama F-ölçüsü değeri 86,9 olarak bulunmuştur.

MLP metodu ile yapılan çalışmalarda Çizelge 4.4 ve Çizelge 4.5'den elde edilen sonuçlara göre gövdelerde en yüksek değerler gövde 1-gramlarda eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 96,9, ortalama F-ölçüsü değeri 90,7 olmaktadır. Sözcüklerde en yüksek değerler sözcük 1-gramlarda eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 97, ortalama F-ölçüsü değeri 91 olarak elde edilmiştir. Hecelerde en yüksek değerler hece 1-gramlarda eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 98,4, ortalama F-ölçüsü değeri 95,3 ve karakterlerde en yüksek değerler karakter 4-gramlarda eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 99 ortalama F-ölçüsü değeri 97 olarak bulunmuştur.

SVM metodunun sisteme uygulanması sonucu Çizelge 4.6 ve Çizelge 4.7'dan elde edilen sonuçlara göre gövdelerde en yüksek değerler gövde 1-gramlarda eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 99,8, ortalama F-ölçüsü değeri 99,3 olmaktadır. Sözcüklerde en yüksek değerler sözcük 1-gramlarda eşik değer 0,5 olarak alındığında ortalama doğruluk değeri 99,9, ortalama F-ölçüsü değeri 99,7 olarak elde edilmiştir. Hecelerde en yüksek değerler hece 2-gramlarda eşik değer 0,25 olarak alındığında ortalama doğruluk değeri 99,8, ortalama F-ölçüsü değeri 99,3 ve karakterlerde en yüksek değerler karakter 4-gramlarda eşik değer

0,25 olarak alındığında ortalama doğruluk değeri 99,8 ortalama F-ölçüsü değeri 99,3 olarak bulunmuştur.

Başarı oranlarının karşılaştırılmasında genel olarak en yüksek başarılar eşik değeri 0,25 alındığında bulunmaktadır. Her 3 yöntem de karşılaştırıldığında en yüksek başarı oranı SVM metoduyla elde edilmiştir. Gövde n-gram ve sözcük n-gramlarda n değeri 1 alındığında en yüksek başarı değerleri elde edilmiş olup n değeri arttığında başarı oranının düştüğü gözlemlenmiştir. Hece n-gramlarda ise en yüksek başarı 2-gramlarda olmaktadır. Karakter n-gramları incelendiğinde K-NN metodunda doğruluk oranı 3-gramlara kadar artmış ardından düşmüştür, MLP ve SVM metotlarında en yüksek doğruluk oranı 4-gramlarda elde edilmiş olup n değeri daha yüksek değerler aldığında doğruluk oranı düşmektedir.

Sonuç olarak bütün bu oranlar göz önüne alındığında en yüksek başarı oranı sisteme SVM metodunun uygulanmasıyla eşik değer 0,50 olarak alındığında sözcük 1-gramlarda %99,9 olarak elde edilmiştir.

5. TARTIŞMA VE SONUÇLAR

Bu çalışmada, K-NN, MLP, SVM metotları kullanılarak web sayfalarından elde edilen dokümanlar gövde n-gramları, sözcük n-gramları, hece n-gramları ve karakter n-gramları olmak üzere 4 farklı kategoriye göre sınıflandırılmıştır. Öznitelik vektörlerinin oluşturulmasında eşik değer olarak 0,25, 0,50, 0,75 ve 0,90 değerleri alınmıştır. K-NN metodunun sisteme uygulanmasında en yakın k değeri olarak 1,3,5 ve 7 değerleri alınarak sonuçlar karşılaştırılmıştır. MLP metodu ile sistem denendiğinde 200000 iterasyon sonucu ve ayrıca SVM metodu uygulanarak sonuçlar elde edilmiştir.

En yüksek başarılar, genel olarak eşik değeri 0,25 alındığında elde edilmiştir. Gövde tabanlı uygulamada en iyi sonuçlar 1-gramlarda, sözcük tabanlı uygulamada 1-gramlarda, hece tabanlı uygulamalarda MLP ve SVM'de 2-gramlarda K-NN'de 1-gramlarda, karakter tabanlı uygulamalarda MLP ve SVM'de 4-gramlarda K-NN'de 3-gramlarda elde edilmiştir. Karşılaştırma kıstası olarak ortalama doğruluk değerleri ve F-ölçüsü değerleri kullanılmıştır.

Her 3 yöntem de karşılaştırıldığında en yüksek başarı oranı sisteme SVM metodunun uygulanmasıyla eşik değer 0,50 olarak alındığında sözcük 1-gramlarda ortalama doğruluk değeri %99,9 ve ortalama F-ölçüsü değeri de %99,7 olarak elde edilmiştir.

Öznitelik vektör uzayının oluşturulmasında kullanılan eşik değeri, K-NN metodunda kullanılan k değerleri, MLP metodunda kullanılan öğrenme katsayısı, iterasyon sayısı, kullanılan aktivasyon fonksiyonu, eğitim aşamasında sisteme verilen dokümanların sayısı gibi faktörler sistemin doğruluk oranına etki eder. Daha önce yapılan çalışmalarda MLP metodu kullanılarak sonuçlar elde edilmiştir(Kolyiğit vd.,2012; Yılmaz vd., 2012). Bu çalışmada 3 farklı yöntem denenmiştir fakat başka metotlar kullanılarak test işlemine göre başarı oranları karşılaştırılabilir.

KAYNAKLAR

Adsett, C. R., Marchand Y., Keselj V. 2009. Syllabification rules versus data-driven methods in a language with low syllabic comlexity: The case of Italian. **Computer Speech and Language**, 23(4): 444-463.

Akın, A.A., Akın, M.D. 2010. Zemberek an open source NLP framework for Turkish Languages. Technical Report, http://zemberek.gogle.com/files/zemberek_makale.pdf. Accessed

Aşlıyan, R., Günel, K. 2011. A comparison of syllabifying algorithms for Turkish, **Journal of Advanced Research in Computer Science**, 3(1):58-78.

Cebiroğlu, G., Adalı, E. 2002. Root Reaching Method Without Dictionary. Istanbul Technical University Computer Engineering Department, Istanbul, Turkey

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features**. Proceedings of The 10th European Conference on Machine Learning (ECML-98) Nedellec, C., Rouveirol, C., Eds.**, pp 137–142, Heidelberg et al.

Kazama, J., Tsujii, J. 2005. Maximum entropy models with inequality constraints: A case study on text categorization. **Mach. Learn.**, 60(1–3):159–194.

Kıyak, E. 2003. Bulanık Mantık Yöntemiyle Uçuş Kontrol Uygulamaları, Anadolu Üniversitesi, Sivil Havacılık Y.O., Yüksek Lisans Tezi, Eskişehir.

Kim, S.B., Rim, H.C., Yook, D., Lim, H.S. 2002. Effective methods for improving naive bayes text classifiers, **In The 7th Pacific Rim International Conference on Artificial Intelligence**, pp. 414–423., Springer-Verlag Berlin Heidelberg.

Kolyiğit Ö., Aşlıyan R., Günel K. 2012. Türkçe Dokümanlar İçin Yazar Tanıma, Akademik Bilişim Konferansı, Uşak.

Köksal, A. 1975. Automatic Morbhological Analysis of Turkish. Hacettepe University, Ankara, Turkey

Kut, A., Alpkoçak, A., Özkarahan, E. 1995. Bilgi Bulma Sistemleri İçin Otomatik Türkçe Dizinleme Yöntemi. Bilişim Bildirileri, Dokuz Eylül University, İzmir, Turkey

Li, R., Wang, J., Chen, X., Tao, X., Hu, Y. 2005. Using maximum entropy model for chinese text categorization. **J. Comput. Res. Dev.**, 42(1):94–101.

Liu, W.Y., Song, N. 2003. A fuzzy approach to classification of text documents. **J. Comput. Sci. Technol.**, 18(5):640–647.

Marchand, Y., Adsett, C. R., Damper, R. I. 2009. Automatic syllabification in English: A comparison of different algorithms**. Language and Speech**, 52(5): 1-27.

Oflazer, K. 1999. Dependency Parsing With an Extended Finite State Approach. Department of Computer Engineering, Bilkent University, Ankara, Turkey.

Özkan, Y. 2008. Karar Ağaçları ile Sınıflandırma. Veri Madenciliği Yöntemleri Dr. Rifat Çölkesen Dr. Cengiz Uğutkaya Papatya Yayıncılık, İstanbul, 216p.

Öztemel, E. 2003. Yapay Sinir Ağı Modeli (Öğretmenli Öğrenme) Çok Katamanlı Algılayıcı. Yapay Sinir Ağları Papatya Yayıncılık, İstanbul, 232p.

Pilavcılar, İ., S. 2007. Metin Madenciliği İle Metin Sınıflandırma, Yıldız Teknik Üniversitesi Matematik Mühendisliği, Yüksek Lisans Tezi, İstanbul.

Solak, A., Can, F. 1994. Effects of Stemming On Turkish Text Retrieval. Departman of Computer Engineering and İnformation Sciences, Bilkent University, Ankara, Turkey.

Solak, A., Oflazer, K. 1993. Design and Implementation of a Spelling Checker for Turkish. Department of Computer Engineering and Information Science, Bilkent University, Ankara, Turkey.

Soucy, P., Mineau, G.W. 2001. A simple K-NN algorithm for text categorization**. In Proceeding of The First IEEE International Conference On Data Mining (ICDM_01)**, pp 647–648. IEEE Computer Society Washington, DC, USA.

Ucoluk, G., Toroslu, I. H. 1997. A genetic algorithm approach for verification of the syllable based text compression technique. **Journal of Information Science**, 23(5): 365-372.

Wu, M.C., Lin, S.Y., Lin, C.H. 2006. An effective application of decision tree to stock trading. **Expert Syst Appl**, 31(2):270–274.

Yang, Y., Liu, X. 1999. A re-examination of text categorization methods. **Proceedings Of SIGIR'99**, pp 42–49., Berkley, CA USA.

Yılmaz R., Aşlıyan R. ve Günel K. 2012. Otomatik Doküman Sınıflandırma, Akademik Bilişim konferansı, Uşak.

Yılmaz, S. 2006. KOÜ, Bulanık Mantık ve Mühendislik Uygulamaları Ders Notları.

EKLER

Ek 1. Sözcük frekansını hesaplayan program

```
clc
clear all
fid = fopen('birlestirilmis_dosyalar\egitim.txt');
say=1;
while  ~feof(fid)
 egitim_sozcukler{say} = fgetl(fid);
 egitim_frekanslar{say} = str2num(fgetl(fid));
  say=say+1;
end
save 'MatDosyalar\egitim_sozcukler.mat' egitim_sozcukler;
save 'MatDosyalar\egitim_frekanslar.mat' egitim_frekanslar;
fprintf('Egitim sozcukler ve frekanslar mat dosyalari olusturuldu.\n');
fclose(fid);
 clear all
fid = fopen('birlestirilmis_dosyalar\ekonomi.txt');
say=1;
while  ~feof(fid)
 ekonomi_sozcukler{say} = fgetl(fid);
 ekonomi_frekanslar{say} = str2num(fgetl(fid));
  say=say+1;
end
save 'MatDosyalar\ekonomi_sozcukler.mat' ekonomi_sozcukler;
save 'MatDosyalar\ekonomi_frekanslar.mat' ekonomi_frekanslar;
fprintf('Ekonomi sozcukler ve frekanslar mat dosyalari olusturuldu.\n');
```

```
fclose(fid);
clear all
fid = fopen('birlestirilmis_dosyalar\kultur_sanat.txt');
say=1;
while  ~feof(fid)
 kultur_sanat_sozcukler{say} = fgetl(fid);
 kultur_sanat_frekanslar{say} = str2num(fgetl(fid));
  say=say+1;
end
save 'MatDosyalar\kultur_sanat_sozcukler.mat' kultur_sanat_sozcukler;
save 'MatDosyalar\kultur_sanat_frekanslar.mat' kultur_sanat_frekanslar;
fprintf('Kultur sanat sozcukler ve frekanslar mat dosyalari olusturuldu.\n');
fclose(fid);
clear all
fid = fopen('birlestirilmis_dosyalar\otomobil.txt');
say=1;
while  ~feof(fid)
 otomobil_sozcukler{say} = fgetl(fid);
 otomobil_frekanslar{say} = str2num(fgetl(fid));
  say=say+1;
end
save 'MatDosyalar\otomobil_sozcukler.mat' otomobil_sozcukler;
save 'MatDosyalar\otomobil_frekanslar.mat' otomobil_frekanslar;
fprintf('Otomobil sozcukler ve frekanslar mat dosyalari olusturuldu.\n');
fclose(fid);
clear all
```

```
fid = fopen('birlestirilmis_dosyalar\saglik.txt');
say=1;
while  ~feof(fid)
saglik_sozcukler{say} = fgetl(fid);
 saglik_frekanslar{say} = str2num(fgetl(fid));
   say=say+1;
end
save 'MatDosyalar\saglik_sozcukler.mat' saglik_sozcukler;
save 'MatDosyalar\saglik_frekanslar.mat' saglik_frekanslar;
fprintf('Saglik sozcukler ve frekanslar mat dosyalari olusturuldu.\n');
fclose(fid);
clear all
fid = fopen('birlestirilmis_dosyalar\spor.txt');
say=1;
while  ~feof(fid)
 spor_sozcukler{say} = fgetl(fid);
 spor_frekanslar{say} = str2num(fgetl(fid));
  say=say+1;
end
save 'MatDosyalar\spor_sozcukler.mat' spor_sozcukler;
save 'MatDosyalar\spor_frekanslar.mat' spor_frekanslar;
fprintf('Spor sozcukler ve frekanslar mat dosyalari olusturuldu.\n');
fclose(fid);
fprintf('\n Islem Tamam...\n');
```

Ek 2. Öznitelik vektör uzayını oluşturan program

```
clear all
clc
load MatDosyalar\saglik_sozcukler.mat;
load MatDosyalar\egitim_sozcukler.mat;
load MatDosyalar\spor_sozcukler.mat;
load MatDosyalar\ekonomi_sozcukler.mat;
load MatDosyalar\kultur_sanat_sozcukler.mat;
load MatDosyalar\otomobil_sozcukler.mat;
load MatDosyalar\saglik_frekanslar.mat;
load MatDosyalar\egitim_frekanslar.mat;
load MatDosyalar\spor_frekanslar.mat;
load MatDosyalar\ekonomi_frekanslar.mat;
load MatDosyalar\kultur_sanat_frekanslar.mat;
load MatDosyalar\otomobil_frekanslar.mat;
saglik_toplam=0;
for i=1:length(saglik_sozcukler)
    saglik_toplam=saglik_toplam+saglik_frekanslar{i};
end
for i=1:length(saglik_sozcukler)
    saglik_frekanslar{i}= saglik_frekanslar{i}/saglik_toplam;
end
 egitim_toplam=0;
for i=1:length(egitim_sozcukler)
    egitim_toplam=egitim_toplam+egitim_frekanslar{i};
end
```

```
for i=1:length(egitim_sozcukler)
   egitim_frekanslar{i}= egitim_frekanslar{i}/egitim_toplam;
end
  spor_toplam=0;
for i=1:length(spor_sozcukler)
    spor_toplam=spor_toplam+spor_frekanslar{i};
end
for i=1:length(spor_sozcukler)
   spor_frekanslar{i}= spor_frekanslar{i}/spor_toplam;
end
 ekonomi_toplam=0;
for i=1:length(ekonomi_sozcukler)
    ekonomi_toplam=ekonomi_toplam+ekonomi_frekanslar{i};
end
for i=1:length(ekonomi_sozcukler)
    ekonomi_frekanslar{i}= ekonomi_frekanslar{i}/ekonomi_toplam;
end
kultur_sanat_toplam=0;
for i=1:length(kultur_sanat_sozcukler)
    kultur_sanat_toplam=kultur_sanat_toplam+kultur_sanat_frekanslar{i};
end
for i=1:length(kultur_sanat_sozcukler)
    kultur_sanat_frekanslar{i}= kultur_sanat_frekanslar{i}/kultur_sanat_toplam;
end
 otomobil_toplam=0;
for i=1:length(otomobil_sozcukler)
```

```
    otomobil_toplam=otomobil_toplam+otomobil_frekanslar{i};
end
for i=1:length(otomobil_sozcukler)
    otomobil_frekanslar{i}= otomobil_frekanslar{i}/otomobil_toplam;
end
esik_deger=0.25;
frekans_esik=0.0002;
saglik_yeni_sozcukler={};
ind=1;
for i=1:length(saglik_sozcukler)
    frekans1=saglik_frekanslar{i};
    frekans2=0;
    for  j=1: length(egitim_sozcukler)
        if (1==strcmp(saglik_sozcukler{i},egitim_sozcukler{j}))
            frekans2=egitim_frekanslar{j};
        end
    end
    frekans3=0;
    for  j=1: length(spor_sozcukler)
        if (1==strcmp(saglik_sozcukler{i},spor_sozcukler{j}))
            frekans3=spor_frekanslar{j};
        end
    end
    frekans4=0;
    for  j=1: length(ekonomi_sozcukler)
        if (1==strcmp(saglik_sozcukler{i},ekonomi_sozcukler{j}))
```

```
            frekans4=ekonomi_frekanslar{j};
        end
    end
    frekans5=0;
    for  j=1: length(kultur_sanat_sozcukler)
        if (1==strcmp(saglik_sozcukler{i},kultur_sanat_sozcukler{j}))
            frekans5=kultur_sanat_frekanslar{j};
        end
    end
    frekans6=0;
    for  j=1: length(otomobil_sozcukler)
        if (1==strcmp(saglik_sozcukler{i},otomobil_sozcukler{j}))
            frekans6=otomobil_frekanslar{j};
        end
    end
    if (frekans1*esik_deger >= frekans2) & (frekans1*esik_deger >= frekans3) &
(frekans1*esik_deger >= frekans4) & (frekans1*esik_deger >= frekans5) &
(frekans1*esik_deger >= frekans6) & (saglik_frekanslar{i}>=frekans_esik)
        saglik_yeni_sozcukler{ind}=saglik_sozcukler{i};
        ind=ind+1;
    end
  end
egitim_yeni_sozcukler={};
ind=1;
for i=1:length(egitim_sozcukler)
    frekans_A=egitim_frekanslar{i};
          frekans_B=0;
```

```
    for j=1:length(saglik_sozcukler)
        if (1==strcmp(egitim_sozcukler{i},saglik_sozcukler(j)))
            frekans_B=saglik_frekanslar{j};
        end
    end
        frekans_C=0;
        for j=1:length(spor_sozcukler)
            if (1==strcmp(egitim_sozcukler{i},spor_sozcukler{j}))
                frekans_C=spor_frekanslar{j};
            end
        end
        frekans_D=0;
        for j=1:length(ekonomi_sozcukler)
            if (1==strcmp(egitim_sozcukler{i},ekonomi_sozcukler{j}))
                frekans_D=ekonomi_frekanslar{j};
            end
        end
        frekans_E=0;
        for j=1:length(kultur_sanat_sozcukler)
            if (1==strcmp(egitim_sozcukler{i},kultur_sanat_sozcukler{j}))
                frekans_E=kultur_sanat_frekanslar{j};
            end
        end
        frekans_F=0;
        for j=1:length(otomobil_sozcukler)
            if (1==strcmp(egitim_sozcukler{i},otomobil_sozcukler{j}))
```

```
            frekans_F=otomobil_frekanslar{j};
          end
        end
        if((frekans_A*esik_deger>=frekans_B) &
(frekans_A*esik_deger>=frekans_C) & (frekans_A*esik_deger>=frekans_D) &
(frekans_A*esik_deger>=frekans_E) & (frekans_A*esik_deger>=frekans_F) &
(egitim_frekanslar{i}>=frekans_esik))
    egitim_yeni_sozcukler{ind}=egitim_sozcukler{i};
    ind=ind+1;
        end
end
spor_yeni_sozcukler={};
ind=1;
for i=1:length(spor_sozcukler)
    frekans_X=spor_frekanslar{i};
    frekans_Y=0;
    for j=1:length(saglik_sozcukler)
      if (1==strcmp(spor_sozcukler{i},saglik_sozcukler{j}))
       frekans_Y=saglik_frekanslar{j};
      end
    end
        frekans_Z=0;
        for j=1:length(egitim_sozcukler)
            if(1==strcmp(spor_sozcukler{i},egitim_sozcukler{j}))
                frekans_Z=egitim_frekanslar{j};
            end
        end
```

```
        frekans_K=0;
        for j=1:length(ekonomi_sozcukler)
            if(1==strcmp(spor_sozcukler{i},ekonomi_sozcukler{j}))
                frekans_K=ekonomi_frekanslar{j};
            end
        end
        frekans_L=0;
        for j=1:length(kultur_sanat_sozcukler)
            if(1==strcmp(spor_sozcukler{i},kultur_sanat_sozcukler{j}))
                frekans_L=kultur_sanat_frekanslar{j};
            end
        end
        frekans_M=0;
        for j=1:length(otomobil_sozcukler)
            if(1==strcmp(spor_sozcukler{i},otomobil_sozcukler{j}))
                frekans_M=otomobil_frekanslar{j};
            end
        end
        if((frekans_X*esik_deger>=frekans_Y) &
(frekans_X*esik_deger>=frekans_Z) & (frekans_X*esik_deger>=frekans_K) &
(frekans_X*esik_deger>=frekans_L) & (frekans_X*esik_deger>=frekans_M) &
(spor_frekanslar{i}>=frekans_esik))
            spor_yeni_sozcukler{ind}=spor_sozcukler{i};
            ind=ind+1;
        end
    end
    ekonomi_yeni_sozcukler={};
```

```
ind=1;
for i=1:length(ekonomi_sozcukler)
   frekans_a=ekonomi_frekanslar{i};
   frekans_b=0;
   for  j=1: length(egitim_sozcukler)
      if (1==strcmp(ekonomi_sozcukler{i},egitim_sozcukler{j}))
         frekans_b=egitim_frekanslar{j};
      end
   end
   frekans_c=0;
   for  j=1: length(saglik_sozcukler)
      if (1==strcmp(ekonomi_sozcukler{i},saglik_sozcukler{j}))
         frekans_c=saglik_frekanslar{j};
      end
   end
   frekans_d=0;
   for  j=1: length(spor_sozcukler)
      if (1==strcmp(ekonomi_sozcukler{i},spor_sozcukler{j}))
         frekans_d=spor_frekanslar{j};
      end
   end
   frekans_e=0;
   for  j=1: length(kultur_sanat_sozcukler)
      if (1==strcmp(ekonomi_sozcukler{i},kultur_sanat_sozcukler{j}))
         frekans_e=kultur_sanat_frekanslar{j};
      end
```

```
    end
    frekans_f=0;
    for  j=1: length(otomobil_sozcukler)
        if (1==strcmp(ekonomi_sozcukler{i},otomobil_sozcukler{j}))
            frekans_f=otomobil_frekanslar{j};
        end
    end
    if (frekans_a*esik_deger >= frekans_b) & (frekans_a*esik_deger >= frekans_c)
& (frekans_a*esik_deger >= frekans_d) & (frekans_a*esik_deger >= frekans_e) &
(frekans_a*esik_deger >= frekans_f) & (ekonomi_frekanslar{i}>=frekans_esik)
        ekonomi_yeni_sozcukler{ind}=ekonomi_sozcukler{i};
        ind=ind+1;
    end
end




kultur_sanat_yeni_sozcukler={};
ind=1;
for i=1:length(kultur_sanat_sozcukler)
    frekans_x=kultur_sanat_frekanslar{i};
    frekans_y=0;
    for  j=1: length(egitim_sozcukler)
        if (1==strcmp(kultur_sanat_sozcukler{i},egitim_sozcukler{j}))
            frekans_y=egitim_frekanslar{j};
        end
    end
```

```
    frekans_z=0;
    for  j=1: length(saglik_sozcukler)
       if (1==strcmp(kultur_sanat_sozcukler{i},saglik_sozcukler{j}))
          frekans_z=saglik_frekanslar{j};
       end
    end
    frekans_k=0;
    for  j=1: length(spor_sozcukler)
       if (1==strcmp(kultur_sanat_sozcukler{i},spor_sozcukler{j}))
          frekans_k=spor_frekanslar{j};
       end
    end
    frekans_l=0;
    for  j=1: length(ekonomi_sozcukler)
       if (1==strcmp(kultur_sanat_sozcukler{i},ekonomi_sozcukler{j}))
          frekans_l=ekonomi_frekanslar{j};
       end
    end
    frekans_m=0;
    for  j=1: length(otomobil_sozcukler)
       if (1==strcmp(kultur_sanat_sozcukler{i},otomobil_sozcukler{j}))
          frekans_m=otomobil_frekanslar{j};
       end
    end
    if (frekans_x*esik_deger >= frekans_y) & (frekans_x*esik_deger >= frekans_z)
& (frekans_x*esik_deger >= frekans_k) & (frekans_x*esik_deger >= frekans_l) &
```

```
(frekans_x*esik_deger >= frekans_m) &
(kultur_sanat_frekanslar{i}>=frekans_esik)
        kultur_sanat_yeni_sozcukler{ind}=kultur_sanat_sozcukler{i};
        ind=ind+1;
    end
end
otomobil_yeni_sozcukler={};
ind=1;
for i=1:length(otomobil_sozcukler)
    frekans_01=otomobil_frekanslar{i};

    frekans_02=0;
    for  j=1: length(egitim_sozcukler)
        if (1==strcmp(otomobil_sozcukler{i},egitim_sozcukler{j}))
            frekans_02=egitim_frekanslar{j};
        end
    end
    frekans_03=0;
    for  j=1: length(saglik_sozcukler)
        if (1==strcmp(otomobil_sozcukler{i},saglik_sozcukler{j}))
            frekans_03=saglik_frekanslar{j};
        end
    end
    frekans_04=0;
    for  j=1: length(spor_sozcukler)
        if (1==strcmp(otomobil_sozcukler{i},spor_sozcukler{j}))
            frekans_04=spor_frekanslar{j};
```

```
        end
    end
    frekans_05=0;
    for  j=1: length(ekonomi_sozcukler)
        if (1==strcmp(otomobil_sozcukler{i},ekonomi_sozcukler{j}))
            frekans_05=ekonomi_frekanslar{j};
        end
    end
    frekans_06=0;
    for  j=1: length(kultur_sanat_sozcukler)
        if (1==strcmp(otomobil_sozcukler{i},kultur_sanat_sozcukler{j}))
            frekans_06=kultur_sanat_frekanslar{j};
        end
    end
    if (frekans_01*esik_deger >= frekans_02) & (frekans_01*esik_deger>=
frekans_03) & (frekans_01*esik_deger >= frekans_04) & (frekans_01*esik_deger
>= frekans_05) & (frekans_01*esik_deger >= frekans_06) &
(otomobil_frekanslar{i}>=frekans_esik)
        otomobil_yeni_sozcukler{ind}=otomobil_sozcukler{i};
        ind=ind+1;
    end
end
saglik_vektor_uzayi=saglik_yeni_sozcukler;
save 'MatDosyalar\saglik_vektor_uzayi.mat' saglik_vektor_uzayi;
egitim_vektor_uzayi=egitim_yeni_sozcukler;
save 'MatDosyalar\egitim_vektor_uzayi.mat' egitim_vektor_uzayi;
spor_vektor_uzayi=spor_yeni_sozcukler;
```

```
save 'MatDosyalar\spor_vektor_uzayi.mat' spor_vektor_uzayi;


ekonomi_vektor_uzayi=ekonomi_yeni_sozcukler;
save 'MatDosyalar\ekonomi_vektor_uzayi.mat' ekonomi_vektor_uzayi;
kultur_sanat_vektor_uzayi=kultur_sanat_yeni_sozcukler;
save 'MatDosyalar\kultur_sanat_vektor_uzayi.mat' kultur_sanat_vektor_uzayi;
otomobil_vektor_uzayi=otomobil_yeni_sozcukler;
save 'MatDosyalar\otomobil_vektor_uzayi.mat' otomobil_vektor_uzayi;
beep
fprintf('Islem tamam...\n');
```

Ek 3. Eğitim matrisini oluşturan program

```
clear all
clc
A1=[];
A2=[];
load MatDosyalar\saglik_vektor_uzayi.mat;
load MatDosyalar\egitim_vektor_uzayi.mat;
load MatDosyalar\spor_vektor_uzayi.mat;
load MatDosyalar\ekonomi_vektor_uzayi.mat;
load MatDosyalar\kultur_sanat_vektor_uzayi.mat;
load MatDosyalar\otomobil_vektor_uzayi.mat;
V=[egitim_vektor_uzayi,ekonomi_vektor_uzayi,kultur_sanat_vektor_uzayi,otomobil_vektor_uzayi,saglik_vektor_uzayi,spor_vektor_uzayi];
save  'MatDosyalar\V.mat' V
siniflar={'sirali_egitim_tum','sirali_ekonomi_tum','sirali_kultur_sanat_tum','sirali_otomobil_tum','sirali_saglik_tum','sirali_spor_tum'};
for w1=1:length(siniflar)
egitimsay=0;
for w2=51:75
clear dosya s2 gecici s s1 say sozcukler frekanslar
egitimsay =egitimsay+1;
fid = fopen(['butun_dosyalar\' siniflar{w1} '\' num2str(w2) '.txt']);
disp(['butun_dosyalar\' siniflar{w1} '\' num2str(w2) '.txt'])
say=1;
while  ~feof(fid)
sozcukler{say} = fgetl(fid);
frekanslar{say} = str2num(fgetl(fid));
```

```
  say=say+1;
end
fclose(fid);
toplamfrekans=0;
for tp=1:length(frekanslar)
 toplamfrekans=toplamfrekans+frekanslar{tp};
end
for norm=1:length(frekanslar)
frekanslar{norm}=frekanslar{norm}/toplamfrekans;
end
for i=1:length(V)
   for j=1:length(sozcukler)
    if (1==strcmp(V{i},sozcukler{j}))
dosyalarfrekansmatris(i,egitimsay)=frekanslar{j};
      break;
    else
       dosyalarfrekansmatris(i,egitimsay)=0;
    end

   end
end
end
A1=[A1 dosyalarfrekansmatris];
end
A2=A1';
for i=1:length(A2(1,:))
  for j=1:length(A2(:,1))
```

```
    A3{j,i}=A2(j,i);
   end
end
uzunluk=size(A3,2)+1;
for k=1:25
   A3{k,uzunluk} ='egitim';
end
     for k=26:50
   A3{k,uzunluk} ='ekonomi';
     end
    for k=51:75
   A3{k,uzunluk} ='kultur_sanat';
    end
    for k=76:100
   A3{k,uzunluk} ='otomobil';
    end
    for k=101:125
   A3{k,uzunluk} ='saglik';
    end
    for k=126:150
   A3{k,uzunluk} ='spor';
   end
  for k=1:uzunluk
   K{1,k} ='X';
end
A=[K;A3];
```

```
save 'MatDosyalar\input.mat' A;
fprintf('Program bitti. input.mat dosyası (A matrisi) olusturuldu.\n');
```

Ek 4. Test matrisini oluşturan program

```
clc
clear all
B1=[];
load MatDosyalar\saglik_vektor_uzayi.mat
load MatDosyalar\egitim_vektor_uzayi.mat;
load MatDosyalar\spor_vektor_uzayi.mat;
load MatDosyalar\ekonomi_vektor_uzayi.mat;
load MatDosyalar\kultur_sanat_vektor_uzayi.mat;
load MatDosyalar\otomobil_vektor_uzayi.mat;
V=[egitim_vektor_uzayi,ekonomi_vektor_uzayi,kultur_sanat_vektor_uzayi,otomobil_vektor_uzayi,saglik_vektor_uzayi,spor_vektor_uzayi];
save  'MatDosyalar\V.mat' V
siniflar={'sirali_egitim_tum','sirali_ekonomi_tum','sirali_kultur_sanat_tum','sirali_otomobil_tum','sirali_saglik_tum','sirali_spor_tum'};
for w1=1:length(siniflar)
for w2=1:50
clear dosya s2 gecici s s1 say sozcukler frekanslar
fid = fopen(['butun_dosyalar\' siniflar{w1} '\' num2str(w2) '.txt']);
disp(['butun_dosyalar\' siniflar{w1} '\' num2str(w2) '.txt'])
say=1;
while  ~feof(fid)
sozcukler{say} = fgetl(fid);
 frekanslar{say} = str2num(fgetl(fid));
   say=say+1;
end
fclose(fid);
```

```
toplamfrekans=0;
for tp=1:length(frekanslar)
  toplamfrekans=toplamfrekans+frekanslar{tp};
end
for norm=1:length(frekanslar)
frekanslar{norm}=frekanslar{norm}/toplamfrekans;
end
 for i=1:length(V)
    for j=1:length(sozcukler)
     if (1==strcmp(V{i},sozcukler{j}))
        dosyalarfrekansmatris(i,w2)=frekanslar{j};
       break;
     else
         dosyalarfrekansmatris(i,w2)=0;
    end

    end
end

end
B1=[B1 dosyalarfrekansmatris];
end
for i=1:length(B1(1,:))
   for j=1:length(B1(:,1))
    B{j,i}=B1(j,i);
   end
```

```
end
save 'MatDosyalar\output.mat' B;
fprintf('Program bitti. Output.mat dosyasi (B Matrisi) olusturuldu..\n');
```

Ek 5. MLP eğitim programı

```
clear all
clc
tic
load MatDosyalar\input.mat;
P=A(2:end,1:end-1)';
for i=1:size(P,1)
    for j=1:size(P,2)
        YeniP(i,j)=P{i,j};
    end
end
load MatDosyalar\egitim_vektor_uzayi.mat;
ozniteliksozcukler.egitim.bas=1;
ozniteliksozcukler.egitim.son=length(egitim_vektor_uzayi);
load MatDosyalar\ekonomi_vektor_uzayi.mat;
ozniteliksozcukler.ekonomi.bas=ozniteliksozcukler.egitim.son + 1;
ozniteliksozcukler.ekonomi.son=ozniteliksozcukler.egitim.son + ...
 length(ekonomi_vektor_uzayi);
load MatDosyalar\kultur_sanat_vektor_uzayi.mat;
ozniteliksozcukler.kultur_sanat.bas=ozniteliksozcukler.ekonomi.son + 1;
ozniteliksozcukler.kultur_sanat.son=ozniteliksozcukler.ekonomi.son + ...
length(kultur_sanat_vektor_uzayi);
load MatDosyalar\otomobil_vektor_uzayi.mat;
ozniteliksozcukler.otomobil.bas=ozniteliksozcukler.kultur_sanat.son + 1;
ozniteliksozcukler.otomobil.son=ozniteliksozcukler.kultur_sanat.son +...
length(otomobil_vektor_uzayi);
load MatDosyalar\saglik_vektor_uzayi.mat;
```

```
oznitelikSozcukler.saglik.bas=oznitelikSozcukler.otomobil.son +1;
oznitelikSozcukler.saglik.son=oznitelikSozcukler.otomobil.son +...
length(saglik_vektor_uzayi);
load MatDosyalar\spor_vektor_uzayi.mat;
oznitelikSozcukler.spor.bas=oznitelikSozcukler.saglik.son +1;
oznitelikSozcukler.spor.son=oznitelikSozcukler.saglik.son +...
length(spor_vektor_uzayi);
goals=3e-2;
epocs=200000;
lrs=0.02;
T(1,1:25)=0;
T(1,26:50)=0;
T(1,51:75)=0;
T(1,76:100)=0;
T(1,101:125)=0;
T(1,126:150)=1;
clear X Y;
X=[YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,26:50) ...
   YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,51:75) ...
   YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,76:100) ...
   YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,101:125) ...
   YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,126:150) ...
   YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,1:25)];
Y=T;
R=size(X, 1);
inputrange=ones(R,2);
inputrange(:,:)=0.04;
```

```
inputrange(:,1)=0;
nn_net_egitim=newff(inputrange,[30,10,1],{'logsig','logsig','logsig'},'traingd');
nn_net_egitim = init(nn_net_egitim);
nn_net_egitim.trainParam.show = 500;
nn_net_egitim.trainParam.lr = lrs;
nn_net_egitim.trainParam.mc = 0.9;
nn_net_egitim.trainParam.epochs = epocs;
nn_net_egitim.trainParam.goal = goals;
[nn_net_egitim,tr]=train(nn_net_egitim,X,Y);
save MatDosyalar\nn_net_egitim.mat nn_net_egitim
clear X Y;
X=[YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,1:25)...
YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,51:75)...
YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,76:100)...
YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,101:125)...
YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,126:150)...
YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,26:50)];
Y=T;
R=size(X, 1);
inputrange=ones(R,2);
inputrange(:,:)=0.04;
inputrange(:,1)=0
nn_net_ekonomi=newff(inputrange,[30,10,1],{'logsig','logsig','logsig'},'traingd');
nn_net_ekonomi = init(nn_net_ekonomi);
nn_net_ekonomi.trainParam.show = 500;
nn_net_ekonomi.trainParam.lr = lrs;
nn_net_ekonomi.trainParam.mc = 0.9;
nn_net_ekonomi.trainParam.epochs = epocs;
```

```
nn_net_ekonomi.trainParam.goal = goals;
[nn_net_ekonomi,tr]=train(nn_net_ekonomi,X,Y);
save MatDosyalar\nn_net_ekonomi.mat nn_net_ekonomi ;
clear X Y;
X=[YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_...
sanat.son,1:25)
YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_sanat.son,...
26:50)
YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_sanat.son,...
76:100)
YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_sanat.son,...
101:125)
YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_sanat.son,...
126:150)
YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_sanat.son,...
51:75)];
Y=T;
R=size(X, 1);
inputrange=ones(R,2);
inputrange(:,:)=0.04;
inputrange(:,1)=0;
nn_net_kultur_sanat=newff(inputrange,[30,10,1],{'logsig','logsig','logsig'},'traingd'
);
nn_net_kultur_sanat = init(nn_net_kultur_sanat);
nn_net_kultur_sanat.trainParam.show = 500;
nn_net_kultur_sanat.trainParam.lr = lrs;
nn_net_kultur_sanat.trainParam.mc = 0.9;
```

```
nn_net_kultur_sanat.trainParam.epochs = epocs;
nn_net_kultur_sanat.trainParam.goal = goals;
[nn_net_kultur_sanat,tr]=train(nn_net_kultur_sanat,X,Y);
save MatDosyalar\nn_net_kultur_sanat.mat nn_net_kultur_sanat ;
clear X Y;
X=[YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,...
1:25)
YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,...
26:50)
YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,...
51:75)
YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,...
101:125)
YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,...
126:150)
YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,76:100)];
Y=T;
R=size(X, 1);
inputrange=ones(R,2);
inputrange(:,:)=0.04;
inputrange(:,1)=0;
nn_net_otomobil=newff(inputrange,[30,10,1],{'logsig','logsig','logsig'},'traingd');
nn_net_otomobil = init(nn_net_otomobil);
nn_net_otomobil.trainParam.show = 500;
nn_net_otomobil.trainParam.lr = lrs;
nn_net_otomobil.trainParam.mc = 0.9;
```

```
nn_net_otomobil.trainParam.epochs = epocs;
nn_net_otomobil.trainParam.goal = goals;
[nn_net_otomobil,tr]=train(nn_net_otomobil,X,Y);
save MatDosyalar\nn_net_otomobil.mat nn_net_otomobil ;
clear X Y;
X=[YeniP(ozniteliksozcukler.saglik.bas:ozniteliksozcukler.saglik.son,1:25) ...
   YeniP(ozniteliksozcukler.saglik.bas:ozniteliksozcukler.saglik.son,26:50) ...
   YeniP(ozniteliksozcukler.saglik.bas:ozniteliksozcukler.saglik.son,51:75) ...
   YeniP(ozniteliksozcukler.saglik.bas:ozniteliksozcukler.saglik.son,76:100) ...
   YeniP(ozniteliksozcukler.saglik.bas:ozniteliksozcukler.saglik.son,126:150) ...
   YeniP(ozniteliksozcukler.saglik.bas:ozniteliksozcukler.saglik.son,101:125)];
Y=T;
R=size(X, 1);
inputrange=ones(R,2);
inputrange(:,:)=0.04;
inputrange(:,1)=0;
nn_net_saglik=newff(inputrange,[30,10,1],{'logsig','logsig','logsig'},'traingd');
nn_net_saglik = init(nn_net_saglik);
nn_net_saglik.trainParam.show = 500;
nn_net_saglik.trainParam.lr = lrs;
nn_net_saglik.trainParam.mc = 0.9;
nn_net_saglik.trainParam.epochs = epocs;
nn_net_saglik.trainParam.goal = goals;
[nn_net_saglik,tr]=train(nn_net_saglik,X,Y);
save MatDosyalar\nn_net_saglik.mat nn_net_saglik ;
clear X Y;
X=[YeniP(ozniteliksozcukler.spor.bas:ozniteliksozcukler.spor.son,1:25) ...
```

```
    YeniP(oznitelikSozcukler.spor.bas:oznitelikSozcukler.spor.son,26:50) ...
    YeniP(oznitelikSozcukler.spor.bas:oznitelikSozcukler.spor.son,51:75) ...
    YeniP(oznitelikSozcukler.spor.bas:oznitelikSozcukler.spor.son,76:100) ...
    YeniP(oznitelikSozcukler.spor.bas:oznitelikSozcukler.spor.son,101:125) ...
    YeniP(oznitelikSozcukler.spor.bas:oznitelikSozcukler.spor.son,126:150)];
Y=T;
R=size(X, 1);
inputrange=ones(R,2);
inputrange(:,:)=0.04;
inputrange(:,1)=0
nn_net_spor=newff(inputrange,[30,10,1],{'logsig','logsig','logsig'},'traingd');
nn_net_spor = init(nn_net_spor);
nn_net_spor.trainParam.show = 500;
nn_net_spor.trainParam.lr = lrs;
nn_net_spor.trainParam.mc = 0.9;
nn_net_spor.trainParam.epochs = epocs;
nn_net_spor.trainParam.goal = goals;
[nn_net_spor,tr]=train(nn_net_spor,X,Y);
save MatDosyalar\nn_net_spor.mat nn_net_spor ;
toc
```

Ek 6. MLP test programı

```
clear all
clc
load MatDosyalar\output.mat;
P=B;
 for i=1:size(P,1)
    for j=1:size(P,2)
        YeniP(i,j)=P{i,j};
    end
end
load MatDosyalar\egitim_vektor_uzayi.mat;
oznitelikSozcukler.egitim.bas=1;
oznitelikSozcukler.egitim.son=length(egitim_vektor_uzayi);
load MatDosyalar\ekonomi_vektor_uzayi.mat;
oznitelikSozcukler.ekonomi.bas=oznitelikSozcukler.egitim.son + 1;
oznitelikSozcukler.ekonomi.son=oznitelikSozcukler.egitim.son +...
length(ekonomi_vektor_uzayi);
load MatDosyalar\kultur_sanat_vektor_uzayi.mat;
oznitelikSozcukler.kultur_sanat.bas=oznitelikSozcukler.ekonomi.son + 1;
oznitelikSozcukler.kultur_sanat.son=oznitelikSozcukler.ekonomi.son +...
length(kultur_sanat_vektor_uzayi);
load MatDosyalar\otomobil_vektor_uzayi.mat;
oznitelikSozcukler.otomobil.bas=oznitelikSozcukler.kultur_sanat.son + 1;
oznitelikSozcukler.otomobil.son=oznitelikSozcukler.kultur_sanat.son +...
length(otomobil_vektor_uzayi);
load MatDosyalar\saglik_vektor_uzayi.mat;
oznitelikSozcukler.saglik.bas=oznitelikSozcukler.otomobil.son +1;
```

```
oznitelikSozcukler.saglik.son=oznitelikSozcukler.otomobil.son +...
length(saglik_vektor_uzayi);
load MatDosyalar\spor_vektor_uzayi.mat;
oznitelikSozcukler.spor.bas=oznitelikSozcukler.saglik.son +1;
oznitelikSozcukler.spor.son=oznitelikSozcukler.saglik.son +...
length(spor_vektor_uzayi);
sinif_isimleri={'egitim','ekonomi','kultur_sanat','otomobil','saglik','spor'}
 load MatDosyalar\nn_net_egitim.mat;
 load MatDosyalar\nn_net_ekonomi.mat;
 load MatDosyalar\nn_net_kultur_sanat.mat;
 load MatDosyalar\nn_net_otomobil.mat;
 load MatDosyalar\nn_net_saglik.mat;
 load MatDosyalar\nn_net_spor.mat;
 no=1;
for i=1:size(B,2)
   clear dosya fid sinif  oznitelik_inputfrekans sinif_oznitelik_inputfrekans P ;


   clear X;
 X=YeniP(oznitelikSozcukler.egitim.bas:oznitelikSozcukler.egitim.son,i);
    Sonuc= sim(nn_net_egitim, X);
    sonuc1= Sonuc;
    clear X;
X=YeniP(oznitelikSozcukler.ekonomi.bas:oznitelikSozcukler.ekonomi.son,i);
    Sonuc = sim(nn_net_ekonomi, X);
    sonuc2= Sonuc;
    clear X;
```

```
X=YeniP(oznitelikSozcukler.kultur_sanat.bas:oznitelikSozcukler.kultur_sanat.son,i
);
    Sonuc= sim(nn_net_kultur_sanat, X);
    sonuc3= Sonuc;
    clear X;
X=YeniP(oznitelikSozcukler.otomobil.bas:oznitelikSozcukler.otomobil.son,i);
    Sonuc = sim(nn_net_otomobil, X);
    sonuc4= Sonuc;
    clear X;
X=YeniP(oznitelikSozcukler.saglik.bas:oznitelikSozcukler.saglik.son,i);
    Sonuc = sim(nn_net_saglik, X);
    sonuc5= Sonuc;
    clear X;
X=YeniP(oznitelikSozcukler.spor.bas:oznitelikSozcukler.spor.son,i);
    Sonuc = sim(nn_net_spor, X);
    sonuc6= Sonuc;
    [x,sonuc]=max([sonuc1 sonuc2 sonuc3 sonuc4 sonuc5 sonuc6]);
    if no>50
      no=1;
    end
    if i<=50
        sinif='egitim';
        sonucmatris(1,no)=sonuc;
        no=no+1;
    elseif i<=100
        sinif='ekonomi';
        sonucmatris(2,no)=sonuc;
```

```
        no=no+1;
    elseif i<=150
        sinif='kultur_sanat';
        sonucmatris(3,no)=sonuc;
        no=no+1;
    elseif i<=200
        sinif='otomobil';
        sonucmatris(4,no)=sonuc;
        no=no+1;
    elseif i<=250
        sinif='saglik';
        sonucmatris(5,no)=sonuc;
        no=no+1;
    else
        sinif='spor';
        sonucmatris(6,no)=sonuc;
        no=no+1;
    end
end
Evaluate(sonucmatris,sinif_isimleri);
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Rumeysa YILMAZ

Doğum Yeri ve Tarihi : AYDIN-03. 06. 1987

EĞİTİM DURUMU

Lisans Öğrenimi : Afyon Kocatepe Üniversitesi Uşak Fen Edebiyat Fakültesi-Matematik Bölümü

Yüksek Lisans Öğrenimi : Adnan Menderes Üniversitesi-Matematik A.B.D.

Bildiği Yabancı Diller : İngilizce

BİLİMSEL FAALİYETLERİ

Bildiriler : Yılmaz R., Aşlıyan R. ve Günel K. 2012. Otomatik Doküman Sınıflandırma, Akademik Bilişim 2012.

Katıldığı Projeler : Aşlıyan R., Günel K.,Yılmaz R., Kolyiğit Ö., Yıldırım Ö., Web Sayfalarının Sınıflandırılması, Bilimsel Araştırma Projeleri Kurulu, ADU, FEF11007, Aydın, Türkiye, 29.09.2012.

İLETİŞİM

E-posta Adresi : rumeysa2903@gmail.com

Tarih : 04.02.2013