**T.C.**
**İSTANBUL AYDIN ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

# DOĞAL DİL İŞLEME YÖNTEMLERİYLE TÜRKÇE SOSYAL MEDYA VERİLERİ ÜZERİNDE DUYGU ANALİZİ

**YÜKSEK LİSANS TEZİ**

**İLKAY YELMEN**
**(Y1413.010045)**

**Bilgisayar Mühendisliği Ana Bilim Dalı**

**Bilgisayar Mühendisliği Programı**

**Tez Danışmanı: Yrd. Doç. Dr. Metin ZONTUL**

**AĞUSTOS 2016**

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

## Yüksek Lisans Tez Onay Belgesi

Enstitümüz Bilgisayar Mühendisliği Ana Bilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı **Y1413.010045** numaralı öğrencisi **İlkay YELMEN**'ın **"DOĞAL DİL İŞLEME YÖNTEMLERİYLE TÜRKÇE SOSYAL MEDYA VERİLERİ ÜZERİNDE DUYGU ANALİZİ"** adlı tez çalışması Enstitümüz Yönetim Kurulunun 19.07.2016 tarih ve 2016/19 sayılı kararıyla oluşturulan jüri tarafından oybirliği ile Tezli Yüksek Lisans tezi olarak kabul....edilmiştir.

| **Öğretim Üyesi Adı Soyadı** | **İmzası** |
|---|---|
| **Tez Savunma Tarihi :02.08.2016** | |
| **1)Tez Danışmanı: Yrd. Doç. Dr. Metin ZONTUL** | ..... |
| **2) Jüri Üyesi : Yrd. Doç. Dr. Ferdi SÖNMEZ** | ..... |
| **3) Jüri Üyesi : Prof. Dr. Ali GÜNEŞ** | ..... |

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

## YEMİN METNİ

Yüksek Lisans tezi olarak sunduğum "Doğal Dil İşleme Yöntemleriyle Türkçe Sosyal Medya Verileri Üzerinde Duygu Analizi" adlı çalışmanın, tezin proje safhasından sonuçlanmasına kadarki bütün süreçlerde bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografya'da gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve onurumla beyan ederim. (02/08/2016)

İlkay YELMEN

## ÖNSÖZ

Yüksek lisans tezim boyunca benden yardımlarını esirgemeyen, karşılaştığım sorunlarda bilgi ve deneyimlerini benimle paylaşan değerli hocam ve tez danışmanım Sayın Yrd. Doç. Dr. Metin ZONTUL'a, çalışmamda destekleri olan Sayın Doç. Dr. Oğuz KAYNAR'a ve 2210-C Öncelikli Alanlara Yönelik Yurt İçi Yüksek Lisans Burs Programı kapsamında almış olduğum desteklerden dolayı TÜBİTAK'a teşekkürü bir borç bilirim.
Son olarak, hayatımın her döneminde olduğu gibi tez çalışmalarım boyunca da destek olan aileme sonsuz teşekkür eder, saygılarımı sunarım.

Ağustos 2016

İlkay YELMEN
(Yazılım Mühendisi)

**İÇİNDEKİLER**

**Sayfa**

**KISALTMALAR**

**DA** **:** Duygu Analizi
**DVM** **:** Destek Vektör Makineleri
**DDİ** **:** Doğal Dil İşleme
**MÖ** **:** Makine Öğrenimi
**DÖ** **:** Derin Öğrenme
**KDM** **:** Karar Destek Makinaları
**GA** **:** Genetik Algoritma

## ÇİZELGE LİSTESİ

**SAYFA**

## ŞEKİL LİSTESİ

**Sayfa**

# DOĞAL DİL İŞLEME YÖNTEMLERİYLE TÜRKÇE SOSYAL MEDYA VERİLERİ ÜZERİNDE DUYGU ANALİZİ

## ÖZET

İnternetin sürekli olarak gelişmesi ve hayatımızın vazgeçilmesi olması ile beraber birtakım sosyal paylaşım siteleri ortaya çıkmıştır. İnsanların fikirlerini paylaştığı ve etkileşimde bulunduğu bu sosyal medya platformları veri kaynağı açısından bilim insanlarının adresi olmuştur.

İnsanlar günümüzde istedikleri bilgiye internet üzerinden yaptıkları aramalarla kolaylıkla ulaşabilmektedir. İnternetteki bilgilerin çoğu geribildirime açık olup bu geri bildirimler anketler ve forum siteleri aracılığıyla yeni fikirlerin analizi için toplanmaktadır. Çok fazla internet kullanıcısı olmasından dolayı geri bildirimlerin insan tarafından analiz edilmesi çok zordur. İşte bu noktada duygu analizi kavramı ortaya çıkmıştır. Duygu analizi, metinlerdeki bir konu hakkındaki duygu ve düşüncenin analiz edilerek duygunun pozitif ve negatif olarak sınıflandırılmasını amaçlar.

Öznitelik seçimi sınıflandırma performansı ve başarısını arttırmak için günümüzde sıklıkla kullanılmaktadır. Bu seçimde farklı metotlar kullanılmakta olup amaçlanan veri kümesi içinden sınıflandırmadaki başarıyı etkileyen alakasız niteliklerin devre dışı bırakılıp önemli niteliklerin seçilmesidir. Bu şekilde başarı oranı arttırılabilir.

Bu tez çalışmasında günlük konuşma dili ile yazılan Türkçe metinlerden öznitelik seçimine odaklanılmış olup detaylı ön işlemeden geçen veri üzerinde destek vektör makineleri, yapay sinir ağları ve centroid tabanlı sınıflandırma algoritmaları kullanılmıştır. 3 ayrı GSM operatörünün takipçilerine ait tweetler üzerinde Gini İndeks, Bilgi Kazancı ve Genetik Algoritma 3 farklı sınıflandırma algoritmasıyla hibrit olarak kullanılmıştır. Özellikle boyut indirgemede önemli bir yere sahip olan ve sezgisel olarak çalışan genetik algoritma ile destek vektör makineleri hibrit olarak kullanıldığında 3 farklı GSM operatörü için de %100 başarı elde edilmiştir.

**Anahtar Kelimeler:** Duygu Analizi, Öznitelik Seçimi, Genetik Algoritma, Sosyal Medya, Twitter, Sınıflandırma, Metin Madenciliği

**SENTIMENT ANALYSIS WITH NATURAL LANGUAGE PROCESSING METHODS ON TURKISH SOCIAL MEDIA DATA**

## ABSTRACT

Several social media websites are showed up as Internet's improving continously and becoming an irreplaceable part of our lives. Those sites that people share their opinions and interact with others have become the address of scientists in terms of data source. People can access any information they need easily by doing research on Internet these days. Many of the data are open for feedbacks and these feedbacks are gathered for analyses of new ideas by surveys and forum sites. It is too hard to analyze feedbacks by a person as there are so many Internet users. At this point, emotion analysis concept showed up. Emotion analysis is aimed at classify the emotion as positive and negative by analyze the emotion and thought about a topic in texts.

Entity property selection is used frequently nowadays in order to increase the performance and success in classification. Different methods are used in this selection and it is selecting the important qualities by eliminating the irrelevant features that affect the success in classification in target data set. Thus, hit ratio may increase.

In this thesis, feature selection from Turkish texts written as colloquial is focused and support vector machine, artificial neural networks and centroid based classification algorithms are used on data that has detailed preprocessing. Gini Index, Information Gain and Genetic Algorithm are used as hybrids with 3 different classification algorithms on tweets belonging to 3 different GSM operators' followers. 100% success is achieved for 3 different GSM operators when genetic algorithm, which works as intuitively and has an important role in dimension reduction, and support vector machines are used hybridly.

**Keywords:** Sentiment Analysis, Fature Selection, Genetic Algorithm, Social Media, Twitter, Classification, Text Mining

# 1. GİRİŞ

Sosyal medyanın her geçen gün öneminin artmasıyla rekabet içerisindeki firmalar, alışılagelmiş pazarlama ve satış yöntemleriyle müşteriye erişebilmenin yetersiz olduğunu görmüşler ve ciddi anlamda potansiyeli olan sosyal medyayı gündemlerine almışlardır. Sosyal medya araçlarının ve internetin insanlar tarafından rütin olarak kullandığı görülmekte ve bu durum sosyal medya araçlarının içerisinde oluşan verilerin önemini de artırmaktadır. Bu nedenden ötürü sosyal medya ortamları taşıdığı önemli ve güncel verilerden dolayı birçok sektöre yön vermektedir. Kullanıcıların firmalara ait hizmet ve ürün hakkındaki her türlü görüşleri (olumlu, olumsuz vs.) sosyal medya ortamları aracılığıyla çok hızlı bir şekilde yayılmaktadır.

Sosyal medya araçları arasında bilim insanları tarafından sıkça kullanılan Twitter güncel ve fazla bilgi içermesinden dolayı burada oluşan veriler özellikle doğal dil işleme ve veri madenciliği gibi bilgisayar bilimleri alanında araştırma amaçlı kullanılmakta ve piyasada kullanılabilecek çalışmalar gerçekleştirilmektedir. Örnek olarak salgın hastalıkların önceden tahminlenmesi [1], kullandığımız ilaçların yan etkenlerinin bulunması [2], algının zaman içerisinde farklılaşmasının tahmini [3], turistik bir yere gelen turistlerin atmış oldukları tweetler üzerinde algının analiz edilmesi [4] gibi çalışmalar verilebilir.

2000'li yıllardan önce doğal dil işleme çalışmaları kapsamında duygu sıfatları, öznellik, metaforların yorumlanması, bakış açısı ve etkileri konularını içeren birtakım faaliyetler [5] yapılmış olsa da esas çalışmalar sonrasında yapılmaya başlanmış olup 2003 yılında Duygu Analizi [6] ve Fikir Madenciliği [7] konuları gündeme gelmiştir.

Kelimelerin varlık durumuna göre analizini yapan Elliott[8], Ortony ve ekibi [9] ve ayrıca Stevenson ve ekibi [10] ilkel olarak tabir edebileceğimiz ilk duygu analizi yöntemleriyle fikir madenciliği konusunda çalışmalar yapmıştır.

Eroğul, Türkçe ifadeler üzerinde ilk duygu analizi çalışmalarını gerçekleştirmiş olup İngilizce için kullanılan tekniklerin Türkçe veriler üzerinde uygulanıp uygulanamadığı incelenmiş ve buradaki çalışmalardan yola çıkarak Türkçe'ye uygun yeni metotlar

önerilmiştir. Yapılan çalışmada Türkçe ve İngilizce için sonuçlar çeşitli öznitelikler kullanarak kıyaslanmıştır. Türkçe veriler kullanılarak yapılan deneylerden %86 (F1-ölçümü) olarak en yüksek sonuç elde edilmiştir [11].

Taner, kelimeler arasındaki bağlantılar ve doğal dil işleme yöntemlerini kullanarak özellik tabanlı duygu analizi için bir altyapı oluşturmayı, ayrıca bir ontoloji yapısı kullanıp, kutupluluk bilgisini, alanları ve sonuçları ayrı olarak modellemeyi amaçlamıştır [12].

Albayrak, yapmış olduğu çalışmada psikolojik durumlar ile Türkçe ifadeler ile arasındaki bağlantıyı araştırmıştır. Anksiyeteli, nksiyetesiz, depresyonlu ve depresyonsuz, kişilerden alınan yazılar üzerinde morfolojik analiz uygulanarak elde edilen öznitelikler kullanılarak inceleme yapılmıştır. Çıkan sonuçlar psikolojik durumlar hakkında, Türkçe metinlerde kullanılan kelimelerin önemli seviyede bilgi verdiğini göstermiştir [13].

Akbaş, çalışma kapsamında konu esaslı duygu analizi yapan bir sistem tasarlamıştır. Twitter üzerinden toplanan Türkçe veriler kullanılarak yapılan çalışmada, konulara göre gruplandırılan veriler, Türkçe duygu kelime listesiyle birlikte kelime seçme algoritması uygulanarak, duygu seviyesi belirlenmiş kelimelerin otomatik olarak oluşturulması önerilmiştir [14].

Boynukalın ve Karagöz, yapmış oldukları çalışmada, Türkçe ifadeler üzerinde duygu analizini öfke, üzüntü, korku ve sevinç olarak dört sınıfta ele almışlardır. Türkçe veri seti olmadığı için İngilizce anket cevaplarından oluşan veriler Türkçe'ye çevrilerek ele alınmıştır. Farklı sınıflandırma algoritmaları ile yapılan deneyler sonucunda Türkçe'ye uygun çeşitli yöntemler eklenerek başarılı sonuçlara ulaşılmıştır [15].

Nizam ve Akın, gözetimsiz öğrenme yöntemleri kullanarak Türkçe Twitter verileri üzerinde duygu analizi yapmışlardır. Çalışmada 3 farklı sınıftan oluşan (pozitif, negatif ve nötr) verinin farklı dağılımlar göstermesinin, sınıflandırmadaki başarıya olan etkisi incelenmiş ve yapılan deneyler sonucunda, eşit dağılımlı veri kümesi kullanılarak yapılan sınıflandırmanın dengesiz veri kümesi kullanılarak yapılan sınıflandırmaya göre daha iyi sonuç verdiği tespit edişmiştir [16].

Meriç ve Diri, Twitter verisi kullanarak yapmış oldukları duygu analizi çalışmasında makine öğrenimi metodunu denetimli sınıflandırıcılarla uygulamışlardır. Alan

bağımsız ve bağımlı veri kümelerine uyguladıkları bigram, trigram ve sözcük tabanlı, yaklaşımlarla, bu yaklaşımların ilgili veri kümesi türlerinde denetimli sınıflandırıcılarla elde edilen başarıların kıyaslanması amaçlanmıştır. Çalışma kapsamında karakter n-gram tabanlı denetimli sınıflandırmanın alan bağımlı veri kümelerinde, sözcük tabanlı denetimli sınıflandırmanın ise alan bağımsız veri kümelerinde daha başarılı sonuçlar verdiğini tespit etmişlerdir [17].

Simşek ve Özdemir, yaptıkları çalışma kapsamında Twitter kullanıcılarının ekonomiyle ilgili atmış oldukları tweetler ile borsadaki değişim arasındaki ilişkiyi incelemişlerdir. Sekiz farklı duyguya (hüzün, aşk, öfke, eğlence, iğrenme, sürpriz, korku, utanç) ait 113 özellik seçilip tweetler mutlu ve mutsuz olmak üzere 2 sınıfa ayrılmıştır. Çalışmada borsadaki değişimlerin tweetlerin mutlu-mutsuz olma durumları ile % 45 oranında bağlantılı olduğu tespit edilmiştir [18].

Akba ve arkadaşları, öznitelik seçme yöntemlerini Türkçe film yorumları kullanarak incelemiştir. SVM ve NB algoritmaları kullanılarak yapılan deneylerde iki sınıf için % 83.9, üç sınıf için % 63.3 doğruluk oranı en yüksek olarak SVM'le elde edilmiştir [19].

Sevindi, yapmış olduğu çalışmada film yorumları kullanarak sözlük tabanlı ve makine öğrenimi yaklaşımlarıyla duygu analizi çalışması yapmıştır. Sonuç olarak, makine öğrenimi yaklaşımda SVM'le 0,8258 F-skor değeri, sözlük tabanlı yaklaşımda ise 0,5969 F-skor değeri ortaya çıkmıştır [20].

Özsert ve Özgür, çalışmalarında çoklu dil kullanılmasını önermiştir. İngilizce çekirdek kelimeleri kullanarak farklı diller için pozitif ve negatif çekirdek kelimeleri oluşturacak bir sistem geliştirmişlerdir. Türkçe ve İngilizce dilleri kullanılarak yapılan deneylerde iki dil için de performansın arttığı görülmüştür [21].

Vural, düşünce içerikli web sayfalarının hızlı olarak keşfedilmesini olanak tanıyan düşünce odaklı bir tarayıcı altyapısı önerip deneyler gerçekleştirmiştir. Bu çalışma kapsamında duygu analizi yöntemleri kullanılmıştır [22].

Sosyal medya ortamlarından biriken büyük veri içerisinde duygu barındıran çok miktarda bilgi bulunmaktadır. Bu bilgiyi işleyip öznelliği çıkarmak ve duygu bulunduran metinleri sınıflandırmak, DA'nın en önemli amacıdır. İroni ve iğneleme hem DDİ [23] hem de psikolojide [24,25] büyük öneme sahip olup aynı zamanda da

ilgi çekicidir. Doğal bir metindeki ironi ve iğnelemenin anlaşılması insanlar için bile zor olabilmektedir [24]. İroni ve iğneleme yakalanmasındaki başarı artışı, DA'nın başarımını da büyük oranda artıracağı bilinmektedir.

DA kapsamında birçok bilimsel çalışma yapılmıştır. Bunlardan birçoğu duygu durumu sınıflandırmayla ve başarının artırılması ile ilgilidir. [26,27]. Sınıflandırma için daha çok MÖ ve sözlük tabanlı yaklaşımlar uygulanmaktadır. Özellikle son yıllarda görüntü işleme ve DDİ çalışma alanlarında başarımı yüksek sonuçlar veren Derin Öğrenme (DÖ) yönteminde duygu analizi çalışmaları kapsamında kullanımına başlanmıştır. DÖ yöntemi özellikle İngilizce için fazlaca kullanılan, aynı zamanda da en yüksek başarımların elde edildiği yöntem olarak literatürde kaşımıza çıkmaktadır [28,29].

Jiang ve arkadaşları, tweetler üzerinde hedef-bağımlı bir DA çalışması gerçekleştirmişlerdir. Bu çalışma sonucunda tweetler genel olarak ilgili hedefin yanında başka hedeflerde içerdiğinden dolayı, hedef-bağımlı bir çözümün daha doğru olacağını belirtmişlerdir. Bunun yanı sıra tweetlerin çoğu zaman kısa olmasından kaynaklı olarak ilgili hedef hakkındaki duyguyu yakalamanın zor olduğunu tespit etmişler ve bunun için bağlamında dikkate alınması gerektiğini ifade etmişlerdir. Sınıflandırma çalışmasında linear kernel ile SVM-Light kullanan Jiang ve arkadaşları, % 85.6 oran ile DA alanında önemli bir başarı elde etmişlerdir [30].

Turney, denetimsiz öğrenme algoritması kullanarak anlamsal eğilimlere göre yorumları tavsiye edilebilir veya edilemez olarak sınıflandırmıştır. "Excellent (harika)" ve "poor (kötü)" gibi kelimelerle sınıflandırılmak istenen yorumlardaki kelimelerin ortak bilgilerini kullanıp o yorumların duygusal yönelimlerini tespit etmeye çalışmıştır. Bu çalışmada yeni özellikler oluşturma yeteneğine sahip olan KDM algoritması kullanılmış olup, en iyi sonucu vermiştir [31].

Bo Pang ve Lillian Lee, yaptıkları çalışmada önce veriyi katmanlı sınıflandırma yöntemine göre sınıflandırmış sonrasında öznel bulunanları olumlu veya olumsuz olarak sınıflandırmışlardır. 5000 olumlu, 5000 olumsuz olmak üzere toplamda 1000 yorumun kullanıldığı çalışmada bir önceki çalışmalara göre iki sınıfın kullanıldığı sınıflandırmada %4 lük bir artışla %86 başarı elde edilmiştir [32].

Nguyen ve arkadaşları, twitter verilerini kullanarak önceki tweetlerdeki algıyı kullanıp bu algının zaman içerisinde değişimini inceleyip gelecek tweetlerdeki algıyı tahmin etmeye çalışmışlardır. Modelde en iyi öznitelikler belirlendikten sonra karar ağaçları,

lojistik regresyon ve KDM kullanılmış olup en yüksek başarıyı % 85 oranında KDM vermiştir [33].

Duygu analizinde sınıflandırma yöntemleri üzerinde araştırmaların yanı sıra özellikle son zamanlarda, öznitelik seçimi üzerine olan araştırmalarda artmıştır. Bunun nedeni, veri madenciliği [34,35], tıbbi veri işleme [36] ve dijital bilgi alma [37,38] gibi büyük miktarda veri içeren yeni uygulamaların gün geçtikte çoğalıyor olmasıdır.

Genetik algoritma (GA), öznitelik seçme yöntemlerinde önemli bir yere sahiptir. GA doğal evrim sürecinden esinlenen bir yöntem olup doğal gelişimi taklit eden birçok yapıya sahiptir [39]. Özellikle optimizasyon ve arama problemlerinde büyük potansiyeli vardır. Ayrıca, GA öznitelik seçiminde doğal olarak uygulanabilen bir yöntemdir. Siedlecki ve Sklansky çalışmalarında GA'nın üstün olduğunu klasik algoritmalarla kıyaslayarak ispatlamıştır [40]. Sonrasında, GA'nın öznitelik seçiminde avantajlı olduğunu gösteren farklı çalışmalarda yayınlanmıştır [41,42].

Basit veya tek bir GA ile ilgili sınırlamalar farklı çalışmalarda işlenmiştir. Normal şartlarda, basit bir GA tarafından elde edilen çözümler, klasik sezgisel algoritmalara göre daha iyi sonuçlar elde etmeyebilir. Bu kısıtlamayı aşmak için GA farklı yöntemlerle birleştirilmesi yani hibrit bir yöntem elde edilmesi gerekmektedir. Bunlar, klasik algoritmaların iyi özelliklerini ve özel genetik operatörlerin kullanımını uygulamaya dahil etmektir. Bu prosedürleri izleyen hibrit GA, çeşitli uygulamalara dahil edilmiş olup başarılı sonuçlar elde edilmiştir [43, 44].

Saeys ve arkadaşları, çalışmalarında sınıflandırma için öznitelik seçiminin önemi ve gerekliliğinden bahsetmişlerdir. Ayrıca biyoinformatik alanında sınıflandırma için farklı öznitelik seçme yöntemlerine çalışmalarında değinmişlerdir. Yüksek boyutlu verinin, birçok sınıflandırma algoritması için büyük bir sorun olduğunu, ayrıca bu durumun bellek kullanım ve hesaplama maliyetinin artmasına neden olduğunu söylemişlerdir [45].

Tan ve arkadaşları, boyut indirgeme ile daha anlaşılır modeller elde edip, bu durumun farklı görüntüleme yöntemlerinin kullanımını kolaylaştırması üzerine çalışmalar yapmışlardır [46].

Molina ve arkadaşları, Guyon ve Elisseeff çalışmalarında, sentetik veri kümelerinde ilgili ve ilgisiz nitelikleri değerlendirip bunların etkilerini incelemişlerdir. Üretilen veri setlerini kullanıp yapay yollarla kontrollü deney seti dizayn etmişlerdir [47, 48].

Dash ve Liu, öznitelik seçim sürecini dört adımını anlatmışlar. Bunlar değerlendirme fonksiyonu, doğrulama prosedürü, nesil (üretim) prosedürü ve kriter durdurma olarak ifade edilmektedir. Ayrıca yapılan çalışma kapsamında nesil prosedürü komple, sezgisel ve rastgele olmak üzere üç kategoriye ayrılmıştır. Değerlendirme fonksiyonları ise mesafe, bilgi, bağımlılık, tutarlılık ve sınıflandırıcı hata oranı ölçüleri beş kategoride yer almıştır [49]. Genetik algoritmalar yaygın olarak aynı zamanda örüntü tanıma ve görüntü işleme alanlarında da kullanılmaktadır [50, 51]. Genetik algoritmalar ayrıca birçok farklı özniteliklerin ilişki durumunu tespit etmek ve özniteliklerin bir iyi alt kümesini bulmak için kullanılır [52].

Matsui ve arkadaşları, çalışmalarında beynin gri / beyaz madde bölgelerini sınıflandırmak için yapay sinir ağları (YSA) kullanmışlardır. YSA'nın sınıflandırmadaki performansını artırmak için öznitelik belirlemede GA kullanmışlardır [53].

Literatürdeki Türkçe veriler üzerinde yapılan duygu analizi çalışmaları incelediğinde özellikle günlük konuşma dili ile yazılan metinleri sınıflandırılmanın zor olduğu görülmektedir. Çalışmalarda, öznitelik seçiminden ziyade daha çok sınıflandırma yöntemleri üzerinde odaklanıldığı ve farklı tekniklerle çeşitli sonuçların alındığı tespit edilmiştir. Bu çalışmada, Türkiye'de bulunan 3 farklı GSM operatörü kullanıcılarına ait tweetler toplanarak duygu analizi yapılmıştır. Toplanan veriler ön işlemeden geçirilmiş sonrasında kelime düzeltme, köklerine ayırma ve etiketleme çalışmaları yapılmıştır. İşlemek için kaliteli hale getirilen verilen üzerinde deneysel çalışmalar yapılarak yüksek başarı elde edilmiştir. Bu yüksek başarıyı elde etmek için veri setinin kaliteli hala getirilmesi ve öznitelik seçimi kısımlarına odaklanılmış olup TF, DVM, GA ve TF-IDF, DVM, GA hibrit yöntemleri ayrı uygulandığında %100 başarı elde edilmiştir. Yapılan deneylerde ise boyut indirgemede kullanılan GA'nın diğer öznitelik seçme yöntemlerine göre daha iyi sonuç verdiği görülmüştür.

Tez 5 bölümden oluşmakta olup 2. bölümde doğal dil işleme ve duygu analizi, 3. bölümde öznitelik seçimi, dördüncü bölümde veri madenciliği sınıflandırma

teknikleri, 5. bölümde yapılan deneysel çalışmalar ve son olarak 6. bölümde de sonuç ve önerilere yer verilmiştir.

## 2. DOĞAL DİL İŞLEME

Doğal Dil İşleme (DDİ), bilgisayar bilimlerinin yapay zeka dil bilimi alt kategorisinde bulunan bilgisayar ile gerçek hayatta kullandığımız doğal dillerin etkileşimini inceleyen bilimsel bir çalışma alanıdır. DDİ, doğal dillerin kurallı olan yapısını irdeleyip çözümleyerek işlenip anlaşılması veya yeniden üretilmesi amacını taşımakta olup otomatik çeviri, konuşma, ses tanıma, üretme, ve duygu analizi (DA) gibi birçok konudaki çalışmalarda kullanılmaktadır. Bu çalışma kapsamında ise günlük konuşma dili ile yazılan Türkçe metinler üzerinde doğal dil işleme teknikleri ile ifadeleri anlaşılır hale getirme ve kelimeleri köklerine ayırma gibi işlemler yapılarak metinlerin duygu yönünden otomatik olarak pozitif veya negatif olacak şekilde sınıflandırılması sağlanmıştır.

DDİ alanı ile birlikte metin madenciliğinde bilgi keşfinde daha anlamlı sonuçlar elde edilmeye başlanmıştır. DDİ, bir doğal dili anlama, yorumlama, çözme ve üretme kapasitesi olan bilgisayar sistemlerinin tasarımını yapan bir bilgisayar bilimi alanıdır. DDİ çalışmaları sayesinde insan-bilgisayar etkileşimi belirli bir seviyeye getirilmiş olup yapılan çalışmalardaki başarılar da zaman içerisinde artış göstermektedir [54].

Metin madenciliği çalışmalarında belgelerin analiz edilmesi için, metni içeriğinin anlamını taşıyan kavramların tespit edilmesi önemlidir. Bu kavramlar kelimeler veya kelime gruplarıyla isimlendirilir ve terimler olarak ifade edilir. Belge içerisindeki terimlerin çıkarılması apayrı bir konu olup DDİ çalışmaları kapsamında araştırmaları yapılan bir alandır. DDİ'nin en önemli avantajı belge analizi sürecinde, terimlerin yani kelimelerin ayrıştırılarak eklerinden ayrılıp anlam kaybı oluşmadan en kısa biçimlerine dönüştürülmesidir. Çünkü aynı anlam için kullanılan kelimeler dilbilgisi kurallarından dolayı farklı biçimlerde bulunabilir ve ayrıca bu farklı kullanımlar kaldırılmadığı sürece farklı anlam ifade eden terimler gibi işleme görüp, belgelerin gerçek anlamına ulaşılmasını engel olabilir. DDİ altında yürütülen faaliyetler üç grupta toplanabilir [55]:

- **Biçimbirimsel Çözümleme (Morfolojik Analiz)**

Biçimbirimsel çözümleme, bir cümlede yer alan her kelimenin ayrı ayrı kök ve eklerine ayrılması yani yapısı ile ilgilidir. Sözcüklerin türetilmesi ve ekler Türkçe için oldukça önemlidir. Her dilde biri çekim, diğeri ise türetme olmak üzere 2 farklı sözcük oluşturma yöntemi vardır. Çekim yolu ile sözcük oluşturmada bir sözcüğün farklı şekilleri kullanılır. Türetme ise yeni sözcüklerin, eski sözcüklere yapım ekleri eklenmesi ile oluşturulmasıdır [55, 56, 57]. Kök ve eklere aynı morfem ismide verilmektedir.

Örnek: bitkiler → bitki (kök) + ler (çoğul eki) (bitki ve ler morfemdir)

- **Sözdizimi Çözümlemesi (Sentaktik Çözümleme)**

Sözdizimsel çözümleme, anlamsal analizden önce, cümleyi oluşturan biçimbirimsel öğelerin hiyerarşik kurallara göre olan uyumunu karşılaştırmak kontrol etmektedir. Cümleyi oluşturan geçerli bir yapı olmadığı zaman anlam çıkarımı olmamaktadır. "büyük koş mavi" ifadesi anlamlı bir yapı oluşturmamaktadır [55, 56, 57].

- **Anlam Çözümlemesi (Semantik Çözümlemesi)**

Bir cümlede ne anlatılmak istediğinin anlaşılması, yani istenilen duygu ve düşüncenin ne olduğunun anlaşılması, anlamsal çözümleme ile yapılır.

Anlamsal çözümleme yapılırken, ilk olarak kelimelerin tek tek kontrol edilip veritabanından uygun nesnelerle eşleştirilmesi sağlanmalıdır. Bu süreçte, kelimelerin anlamları daima bir adet olmayabilir. Ayrık kelimelerin bir cümle içindeki doğru anlamını bulmaya "kelime anlam berraklaştırılması" denir. Bu durum, cümle içerisinde geçen bir kelimenin sözlükteki anlamlarının tespit edilerek uygun olanının seçilmesi ile ilgilidir. [55, 56, 57, 58]

## 2.1. Duygu (Sentiment) Analizi

Doğal diller ile yazılmış metinlerin olumlu veya olumsuz durumlarının incelenmesine Duygu (Sentiment) Analizi denilmektedir. Duygu analizi için metinlerin dilbilgisi kurallarına uygun olarak yazılmış olması şarttır. Duygu analizi çalışması bazı doğal dil işleme algoritmaları ve makine öğrenimi yöntemleri ile yapılabilmektedir.

İnsanlar yazarken veya konuşurken 2 farklı kategoride ifade kullanırlar. Bunlar gerçekler ve görüş bildiren ifadelerdir. Gerçekler nesnel ifadeleri, görüşler ise genelde öznel ifadeleri oluşturmaktadır. Görüş anlamsal olarak çok kapsamlı olmasının yanı sıra olay, kişi ve bunların özellikleri hakkında olumlu, olumsuz ve nötr ifadeleri içermektedir [59].

Psikoloji biliminde duygu analizi kavramı ile ilgili kişinin duygusal durumunun kullandığı kelimelere ve bunları kullanma şekliyle bağlantılı olduğu tespit edilmiştir. Bunndan yola çıkarak duygu içeren kelimeler duygu eğilimlerine göre sınıflandırılmış ve taşıdıkları duygu yoğunluğuna göre puanlandırılmıştır [59]. Birçok farklı konu üzerinde uygulaması bulunan duygu analizi özellikle günümüzde yoğun olarak kullanılan sosyal medya araçlarında bilim insanları tarafından büyük ilgi görmektedir.

Mobil cihazların son dönemlerde sıkça kullanılması ve internet erişiminin her yerden sağlanması ile sosyal medya platformları duygu analizi için büyük veri kaynaklarından biri haline gelmiştir. Bilimsel çalışmaların çoğu facebook gibi fonksiyonel detayı fazla olan sosyal medya araçları yerine Twitter gibi metin içeriğinin çok olduğu araçlara yoğunlaşmıştır [60].

Duygu Analizi, metinlerdeki görüş ifade eden kısımları tespit etmek ve bunları farklı açılardan sınıflandırmak amacıyla ortaya çıkmış bir alandır [61]. Duygu Analizi, veri madenciliği, makine öğrenmesi ve doğal dil işleme gibi bilgisayar ve istatistik bilim alanlarında gelişmekte olan, bir araştırma alanıdır. Görüş madenciliği ile ilgili çalışmaları, 9 alt başlıkta incelenmektedir. Bu başlıklar:

1. Öznitelik Seçimi (Feature Selection)
2. Metinden görüş belirten kısımları çıkartma (Subjectivity Extraction)
3. Görüşlerin duygusal bağlamlarını belirleme (Emotion Identification)
4. Görüşlerin kutuplarını belirleme (Identifying Opinion Polarity)
5. Görüşlerin hedefini çıkartma (Target extraction)
6. Görüş Özetleme (Opinion Summarization)
7. Sözlük Oluşturma (Developing Resources)
8. Öğrenme Transferi (Learning Transfer)
9. Görüş Kaynağı Tespiti (Identifying Opinion Source )

Veri madenciliği çalışmalarında yapılacak olan ilk işlem, işlencek veriyi özniteliklerine ayırıp vektörü haline getirmektir. Duygu analizinde genel olarak;

- Kelimeler, ifadeler ve ardışıl kelime / karakter tekrar sıklıkları (n-grams)
- POS (Part of Speech) (Konuşmanın Parçası) Etiketleri
- Bağımlılık Bilgisi
- Kelime altı öznitelikler
- Duygu Simgeleri (Emoticonlar)
- Noktalama İşaretleri
- Büyük / Küçük Harf kullanımları öznitelikleri kullanılmaktadır.

Duygu analizi uygulamalarında;

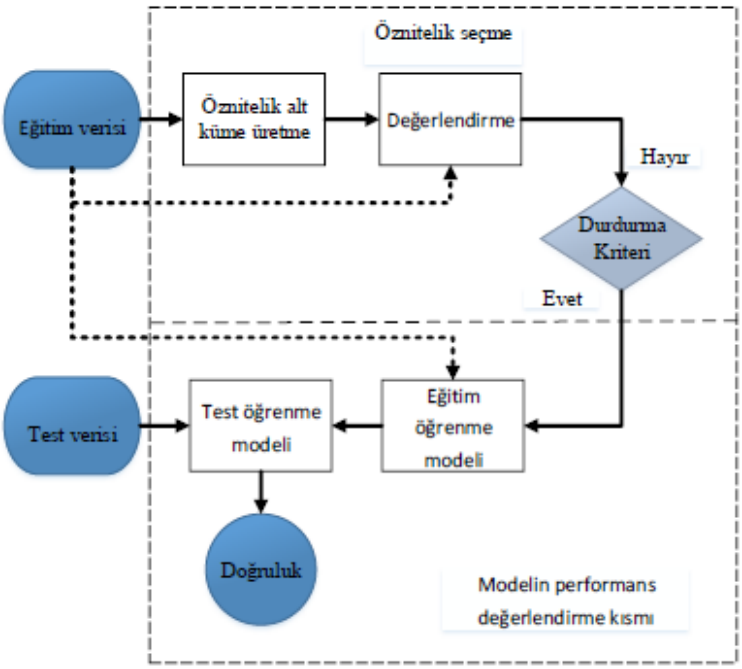
- Sözlük tabanlı
- Denetimli
- Yarı denetimli yaklaşımlar mevcuttur.
- Ayrıca duygu analizi, uygulandığı hedef doğrultusunda da;
- Doküman Seviyesinde
- Cümle Seviyesinde
- Deyim Seviyesinde
- Duygu Seviyesinde sınıflandırılabilir.

Duygu analizi yaklaşımları makine öğrenmesi ve sözlük temelli olarak iki ana dala ayrılabilir. Makine öğrenmesi, denetimli, yarı denetimli ve denetimsiz olmak üzere 3 farklı sınıfta incelenebilir. Sözlük temelli yaklaşımlar ise duygu sözlüğü temelli ve bütünce temelli olarak iki kısımda incelenir [62].

# 3. ÖZNİTELİK SEÇİMİ

Öznitelik seçimi birtakım kriterlere göre orijinal niteliklerden uygun niteliklerin alt kümesini seçmek için geliştirilmiştir. Nitelikleri içeren bir alt kümesinin bulunma nedeni; daha düşük boyutta bir sorunun çözümünün daha kolay olmasıdır. Ayrıca giriş ve çıkış değişkenleri arasında doğrusal olmayan eşleştirmenin anlaşılmasında yol gösterici olur [63]. Öznitelik seçimi, belirli bir büyüklüğe sahip nitelikler kümesinin içerisinden en uygun alt kümenin bulunmasını içermekte olup aynı zamanda da en büyük genellemeyi sağlar [64].

Aşağıda Şekil 3.1'de bir öznitelik seçim süreci anlatılmaktadır. Seçim 3 yönlü olup ilk etapta boyut indirgeme yapılarak sınıflandırıcının önemli bir seviyede tahmin doğruluğu artırabilir. İkinci aşama olarak hesaplama maliyetini düşürür. Birçok öğrenme algoritması eğitim ve tahmin adımlarında, özniteliklerin sayısı fazla olduğu için hesaplama açısından zorlanır. Eğitim algoritmasında, önce öznitelik seçimi adımı hesaplama yükünü azaltabilir. Sonrasında uygulanan boyut indirgeme işlemi, veri üretme sürecine daha iyi bakış açısı sağlar. Bu durum önemlidir, çünkü birçok durumda bilgilendirici öznitelikleri belirtme yeteneği önemlidir [65]. Özniteliklerin seçimi öznitelik sıralama ve öznitelik alt küme seçimi olmak üzere 2 farklı şekilde yapılabilir [67].




**Şekil 3.1:** Bir öznitelik seçim süreci [65]

## 3.1. Bilgi Kazancı (Information Gain)

Bir terimin bir dokümanda olup olmadığına bağlı olarak sınıf belirleme için elde edilen bilginin parça sayısını ölçen terime bilgi kazancı denir.

Bilgi kazancı Shannon 1948 entropisi kullanılarak ölçüldüğünde bilgisel entropi soyut olarak belirli bir bilgi parçasını çözmek için gerekli olan veri parçası sayısıdır [68].

Bilgi kazancının ölçümü öznitelik başlığının kategoride görülüp görülmediği ya da eğer görülüyorsa hangi frekansta görüldüğüne bakılarak yapılır. Daha önceden belirlenmiş değerin bilgi kazancı değerinden yüksek olması durumunda, t öznitelik başlığı öznitelik derlemesinden çıkartılır [69].

Bilgi kazancı hesaplamak için aşağıdaki Shannon'un geliştirdiği entropi hesaplaması kullanılabilir. Eğer terimler yani örnekler aynı sınıfa aitse entropi 0, sınıflar arasında eşit dağılmış ise entropi 1 olacaktır. X sınıfın iyi bir tanımıysa eğer, o özelliğin her bir değerinin sınıf dağılımındaki entropi oranı düşük olacaktır.

$$Entropi = -\sum_{i=1}^{m} p_1 \log_2(p_i) = Bilgi(D) - \sum_{i=1}^{m}(\left(frekans\frac{S_i, D}{|D|}\right) x \log_2(frekans(S_i, D)/|D| \quad \textbf{(3.1)}$$

Formülde D'yi herhangi bir küme olarak düşünürsek, herhangi bir küme için o sınıftaki (S) değerlere göre frekansa bakılır. D kümesini herhangi bir X parçaya böldükten sonra D'yi sınıflandırmak için gerekli olan bilgi:

$$Bilgi_x(D) = \sum_{J=1}^{k}(\left(\frac{|D_j|}{|D|}\right) x\ Bilgi(D_j) \quad \textbf{(3.2)}$$

Bir özniteliğin bilgi kazanımı entropideki düşüş olarak ölçülebilir. Bilgi kazancı veri kütüphanesinde bulunan her doküman için ayrı ayrı hesaplanır. Sonrasında belirli bir değerin altında olan kelimeler topluluktan çıkarılır. Sonuç itibariyle en yüksek kazanım oranı olan öznitelik seçilir. X niteliğine göre bilgi kazanımı aşağıdaki şekilde gösterilmektedir.

$$Kazanım(X) = Bilgi(D) - Bilgi_x(D) \quad \textbf{(3.3)}$$

### 3.2. Gini İndeks

Gini İndeks algoritması, Wenqian Shang ve arkadaşları tarafından, 2007 yılında öznitelik seçimi için Gini İndeks teorisine dayanılarak tanıtılmıştır. Gini İndeks yeni bir ölçme fonksiyonu ile tasarlanmış olup özniteliklerin orijinal öznitelik uzayında seçilmesinde kullanılmıştır. Gini İndeks algoritmasında kirlilik azaldıkça, katkı daha iyi hale gelir. Fakat, Gini İndeks teorisindeki birçok çalışma, ölçü formunda saflığı kabul etmiştir: saflığın değeri arttıkça, katkı daha iyi hale gelir. Bu durumu Shang ve arkadaşlarının Gini İndeks algoritması olarak dikkate alırız [70].

D kümesi n sınıftan örnekler içeriyor ise, Gini İndeks, gini (D) aşağıdaki şekilde ifade edilmektedir.

$$gini\ (D) = 1 - \sum_{J=1}^{n} P_j^2 \quad \textbf{(3.4)}$$

D kümesi A niteliğine göre $D_1$ ve $D_2$ olmak üzere ikiye bölünürse, Gini İndeks (D) aşağıdaki gibi ifade edilir.

$$gini_{A}(D) = \frac{|D_1|}{|D|}gini(D_1) + \frac{|D_2|}{|D|}gini(D_2) \quad \textbf{(3.5)}$$

### 3.3. Genetik Algoritma

Genetik Algoritma (GA) güçlü arama ve optimizasyon tekniği olup Holland tarafından ortaya atılmıştır [71]. GA doğal bir seçim sürecini taklit ederek çözüm uzayında olasılığı yüksek en uygun çözümü bulmaya çalışır. GA, eğitim amacıyla verilen bir uzayın en verimli niteliklerini bulma konusuda oldukça etkilidir.

GA bireylerden oluşan popülasyonunu tekrarlı bir şekilde günceller. Genel anlamda bir başlangıç popülasyon ile başlar ve her tekrarda bir uygunluk fonksiyonuna bağlı olarak bireyleri değerlendirir. Ayrıca yeni popülasyonun, mevcuttakinden daha iyi bir durumda olması garanti edilir. Bazı bireyler değişim göstermeden yeni nesil geçerler. Diğer bireyler üzerinde çaprazlama vemutasyon gibi genetik operatörler uygulanarak çocuklar oluşturulur. GA, durdurma kriterine erişene kadar bu işlemi bir kaç defa tekrarlar. Aşağıda Şekil 3.2'de GA için sözde kod verilmiş olup n, popülasyondaki bireylerin sayısı; $\chi$, her tekrarda çaprazlama ile yerleşen popülasyonun bir kısmı; ve $\mu$ mutasyon oranını temsil etmektedir.

Algoritma: GA ($n$, $\chi$, $\mu$)
//Nesil 0 e ilk değerleri vermek:
$k$: = 0;
$P_k$: $n$'nın bir popülasyonu rastgele-üretilen bireylerin;
//$P_k$ değerlendirilir:
Her $i \epsilon P_k$ için uygunluk ($i$) hesaplanır;
yapmak
{// Nesil oluşturmak $k$ + 1:
// 1. Kopyala:
$P_k$'nın (1-$\chi$) × $n$ üyeleri seçilir; çocuklar $P_{k+1}$ e atılır;
// 2. Çaprazlama:
$P_k$'nın $\chi$ × $n$ üyeleri seçilir, çiftleşir, çocuklar üretilir, çocuklar $P_{k+1}$ e atılır;
// 3. Mutasyon:
$P_{k+1}$'ın $\mu$× $n$ üyeleri seçilir; Her birinde rastgele seçilen bit ters olur;
//$P_{k+1}$ değerlendirilir:
Her $i \epsilon P_k$ için uygunluk ($i$) hesaplanır;
// Artım:
k: = k + 1;
}
$P_k$'de en uygun bireyin uyguluk değeri, yeterli değildir; devam eder
$P_k$ en uygun birey ise döner;

**Şekil 3.2:** GA için sözde kod [72]

Bir popülasyonu bireyler genelde bit dizeleri ile temsil etmektedir. Sonrasında, çaprazlama ve mutasyon kolayca uygulanabilmektedir. Ama bir bit dizisi olarak bireyi kodlayıp, sonra tekrardan çözmek için bazı metotlar uygulanmalıdır.
GA'da bireylerin uygunluğunu test edecek bir fonksiyon bulunmaktadır.

- **Genetik Operatörler**

Yeni neslin bir kısmı $\chi$, ise geri kalanlar çaprazlamayla oluşturulacaktır. Sonra (1-$\chi$) bu nesilden, sonraki nesile direkt kopyalanacaktır. Toplamda, (1 - $\chi$) × n birey kopyalanır.
Aşağıda önemli olan bazı genetik operatörler açıklanmıştır.

**Rulet Tekerleği:** Burada, bireylerin seçilme olasılığı, P (seçim = i) aşağıdaki gibi hesaplanır:

$$P(seçim = i) = \frac{Uygunluk(i)}{\sum_{j=1}^{n} Uygunluk(j)} \quad \textbf{(3.6)}$$

Bir rulet tekerleği her bireyin uygunluk değerine bağlı olarak farklı boyutlarda olabilir. Rulet tekerleği seçimindeki sözde kod Şekil 3.3'te belirtilmiştir [72].

```
Algoritma: RULET TEKERLİĞİ SEÇİMİ ()
r:= rastgele sayı, 0 ≤ r <1;
toplam: = 0;
her i birey için
{       toplam: toplam + P  (seçim = i);
        Eğer r <toplam ise
        {       i e dön;
        }
}
```

**Şekil 3.3:** Rulet tekerleği seçimi için sözde kod [72]

**Çaprazlama:** Populasyondan seçilen iki ebeveynin kromozomlarının bir kısmını yer değiştirmesi ile iki farklı yeni bireyin oluşmasıdır. Çaprazlama işlemi sonucunda populasyonda bulunmayan yeni bireyler oluşur [92]. Çaprazlama operatörü Şekil 3.4'te gösterilmektedir.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Ebeveyn 1 :** | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| **Ebeveyn 2 :** | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| **Çocuk 1 :** | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| **Çocuk 2 :** | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

**Şekil 3.4:** Çaprazlama operatörü

**Mutasyon:** Bir bireyin bir veya birkaç geninin değişerek farklı bir birey haline gelmesidir. Mutasyon populasyonda çeşitliliğin oluşmasını sağlayan operatörlerden biridir [93].

Mutasyon 111010**0**1000 ⟶ 111010**1**1000

Mutasyon içinde mutasyon noktası seçimi, maske oluşturma ve ters çevirme işlemleri bulunmaktadır. Ayrıca düşük bir μ değerine sahip olan mutasyon, birey çeşitliliği açısından da oldukça önemlidir [72].

# 4. VERİ MADENCİLİĞİ

Büyük hacimli olan verilerin içerisinden uygulanabilir, kullanılabilecek ve anlamlı bilgilerin keşfedilmesine veri madenciliği adı verilmektedir. Veri madenciliği veri analizi ile ilgili tekniklerin bütününü oluşturmaktadır [73]. Eldeki verileri kullanarak bir problemi çözmek, veya geleceğe yönelik tahminler yapmak için gerekli bilgileri keşfetmeye yarayan bir araç olan veri madenciliği aynı zamanda veriler arasında gizli kalmış örüntü ortaya çıkaran bir bilgisayar bilimi alanıdır [74]. Durum böyle olunca veri madenciliğinin son zamanlarda popülerliği artmış olup kullanılan en güncel teknolojilerden biri haline getirmiştir. Veri madenciliği kişiler tarafından bilinmeyen bilgileri ortaya çıkardığından dolayı diğer yöntemlerle kıyaslandığında avantajı oldukça fazladır [75].

Veri madenciliği kapsamında denetimli (supervised), yarı denetimli (semi-supervised) ve denetimsiz (unsupervised) olmak üzere 3 farklı öğrenme yöntemi bulunmaktadır. Bu yöntemlerden denetimli öğrenmede veri setindeki tüm veriler sınıf etiketlerinden oluşmaktadır. Yarı denetimli öğrenmede veri seti hem etiketli hem de etiketsiz verileri içermektedir. Denetimsiz öğrenmede ise veri setinde sınıf etiketi bulunmamaktadır. Bu öğrenme yöntemleri ile ilişkili olarak veri madenciliği çalışmaları kapsamında 4 farklı yöntem bulunmaktadır. Bunlar tahminleme (prediction), sınıflandırma (classification), kümeleme (clustering) ve birliktelik kuralları (association rules) dır.

Bu çalışma kapsamında tüm veriler pozitif ve negatif olmak üzere sınıf etiketi taşıdığı için denetimli öğrenme ile birlikte 3 farklı sınıflandırma algoritması (Destek vektör makineleri, yapay sinir ağları, ve centroid tabanlı sınıflandırma) kullanılmıştır.

## 4.1. Yapay Sinir Ağları

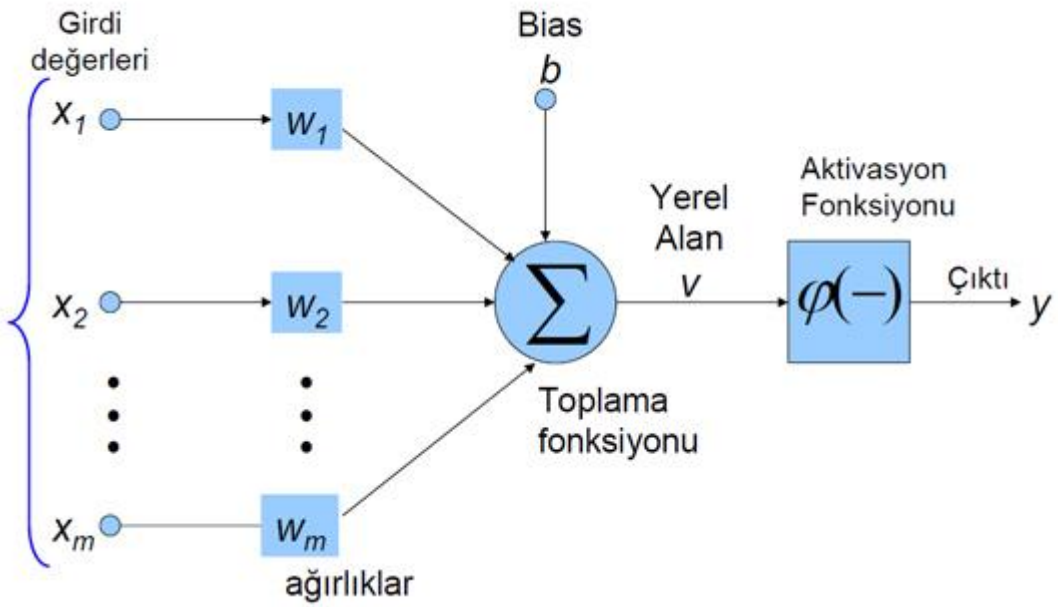
Sinir ağları ile ilgili ilk çalışmalara Donald Hebb tarafından 1949 yılında başlatmıştır. Aynı zamanda Nörolog olan Donald Hebb, beyinle ilgili bir takım çalışmalar yapmış ve beynin birimlerinden biri olan sinir hücresi üzerine bazı incelemeler yapmıştır. Sinir

hücreleri arasındaki ilişkiyi incelemiş ve bu bilgiler ışığında sinir ağı teorisi üzerine çalışmıştır. Yapay sinir ağları kullanılarak yapılan çalışmaların neticesinde çok güçlü ağ ve algoritmalar geliştirilmiştir. Geliştirilen bu yöntemler ile yapılan çalışmalar sonucunda oldukça yüksek başarı oranlarına ulaşıldığı literatürde verilmiştir. [40].

Sinir ağı modellerinin ağ yapısı birbirleriyle bağlantılı olan işlem elemanlarından oluşur. Çıkış işareti isteğe göre değişebildiği gibi ağlar çevre şartlarına göre davranışlarını değiştirebilir.

Yapay sinir ağlarının yapısı göze alındığında birçok hücreden oluştuğu görülür. Bu hücreler eş zamanlı olarak faaliyet gösterir. Bu hücrelerden birisinin işlevini kaybetmesi sistemin çalışmasına engel olmaz. Bu da hataya karışı toleranslı olduklarını gösterir. Yapay sinir ağları makina öğrenmesi gerçekleştirebilirler. Öğrendikleri şeyler hakkında mantıklı kararlar verebilirler. Veriler genel programlamalardaki gibi veri tabanı yada dosyalarda tutulmaz, ağın tamamındaki bağlantılarında saklanmaktadır.

Yapay sinir ağlarının dağıtık yapısından dolayı bilgiler ağ içerisinde dağılmış bir biçimde bulunur. Yani tek bir bağlantı kendi başına bir anlam ifade etmez. Yapay sinir ağlarında veri setiminde bulunan verilerden öğrenme aşaması için eğitim verilerinin belirlenmesi, bu verilerin ağda hedef çıktılara göre ağın eğitilmesi gerekmektedir. İstenen sonuca ulaşabilmek için belirlenen eğitim verilerinin uygun olması önemlidir. YSA'lar eğitimleri esnasında kendilerine verilen örneklerden elde ettikleri genellemeler ile yeni örnekler hakkında bilgi verebilirler [41].




**Şekil 4.1:** Yapay Sinir Hücresi Elemanları

Şekil 4.1'de gösterildiği gibi yapay sinir ağları girdiler, ağırlılar, toplam fonksiyonu, aktivasyon fonksiyonu ve çıktılardan meydana gelmektedir. İşlemci elemana gelen veriler girdilerdir. Girdiler başka bir hücreden yapay sinir hücresine gelebileceği gibi doğrudan dış veri olarakta gelebilir. Gelen veriler, çekirdeğe ulaşmadan önce geldikleri ağların ağırlığıyla çarpılması sonucuyla çekirdeğe iletilirler. Bu olay sayesinden girdilerin üretilecek çıktı üzerindeki etkisinin ayarlanabilmesi sağlanabilir. Ağırlıklar pozitif, negatif ya da sıfır değerlerine sahip olabilir. Yapay sinir hücresinde toplama fonksiyonu, ağırlıklarla çarpılıp gelen girdileri toplayarak o hücrenin girdisini hesaplar [39].

Yapay sinir ağlarında momentum katsayısı bir önceki iterasyonda yapılan değişimin belirli bir yüzdesinin yeni değişim miktarına eklenmesi olarak ifade edilmekte ve öğrenmenin performansını da etkilemektedir. Bu durum sıçrama ile özellikle yerel çözümlere takılan ağların daha iyi sonuçlar elde etmesini sağlamak amacıyla önerilmiştir. Küçük değerler yerel çözümlerden kurtulmayı zorlaştırabilmekte ve çok büyük değerler de çözüme ulaşmada sorunlar çıkartabilmektedir. Yapılan araştırmalar neticesinde momentum katsayısının 0.6 - 0.8 aralığında seçilmesinin daha uygun olacağı belirtilmiştir [76].

Geri yayılım algoritması, basit olması ve uygulamadaki görüş açısı gibi başarılarından dolay bir ağın eğitimi için popüler algoritmalardan biridir [77]. Bu algoritma, hataları çıkıştan girişe doğru azaltmaya (yani geriye doğru) çalışmasından dolayı geri yayılım şeklinde ifade edilmektedir. Geri yayılmalı öğrenme kuralı her bir tabakadaki ağırlıkları ağ çıkışındaki mevcut hata düzeyine göre yeniden hesaplamak amacıyla kullanılmaktadır. Geri yayılımlı modelde giriş, gizli ve çıkış olmak üzere 3 farklı katman bulunmaktadır. Gizli katman sayısını duruma göre artırmak mümkündür.

Çok katmanlı ağlarda geri yayılım algoritması delta kuralı özelinde genelleştirilmiş olup çok katlı ağlarda hesap işlerini öğrenmek için kullanılabilmektedir. Hatalar geri yayılım ağında ileri besleme mekanizması içerisinde kullanılan aynı bağlantılar yardımıyla geriye doğru yayılmaktadır. Bu ağda öğrenme, çift yönlü ve basit hafıza birleştirmeye dayanmaktadır [78].

## 4.2. Destek Vektör Makineleri

Bir makine öğrenme algoritması olan destek vekör makineleri (DVM) öğrenme, sınıflandırma, kümeleme, yoğunluk tahmini ve regresyon kuralları üretmek için kullanılanılır. Lineer olmayan örnek uzayını, örneklerin lineer olarak ayrılabileceği yüksek boyuta aktarıp, farklı örnekler arasındaki maksimum sınırın bulunması esasına dayanır [79].

Sinir ağlarının geliştirilmesi için yedek olarak DVM geliştirildi. DVM mevcut veriye ait sayısal bir model oluştururken, yapay sinir ağları daha sezgisel olarak çalışır. 1960'lı yılların sonlarından 1990'lı yılların başlarına kadar yapılan çalışmalarda, Vapnik tarafından DVM'nin teorik temeli atılmıştır. DVM çok güçlü bir teorik yapıya sahip olmasına karşılık ilk zamanlarda pek taktir görmemiştir. Metin kategorizasyonu, yapay görme ve haneli tanıma ile pratik öğrenme bakımından oldukça iyi hatta mükemmel sonuçlar vermesi üzerine ciddiye alınmaya başlanmıştır. Günümüzde bakıldığında destek vektör makineleri çeşitli problemlerde yapay sinir ağları ve diğer istatistiksel modellere göre çok daha iyi sonuç vermektedir [79].

DVM parametrik olmayan bir modeldir. Fakat parametrik olmama, DVM modelinin tüm parametrelerden yoksun olduğu anlamına gelmez. Bilakis, "öğrenme" (seçimi, tanımlama, tahmin, eğitim ya da ayarlama) burada önemli bir konudur. Klasik

istatiksel çıkarsama modellerinin aksine burada parametreler önceden tanımlı değildir ve bunların sayısı kullanılan eğitim verilerine bağlıdır. Diğer bir bakış açısıyla düşünüldüğünde modelin kapasitesini tanımlayan parametreler veri karmaşıklığı içinde model kapasitesi ile eşleşecek şekilde veri tabanlıdır. Yapısal risk minimizasyonunun temel paradigması olan bu yeni öğrenme modeli Vapnik, Chervonenkis ve diğer bilim adamları tarafından tanıtılmıştır. Yani, iyi bir genelleme özelliğine sahip olması istenen bir model tasarımı yapısal iki temel yaklaşıma sahiptir [80].

Veriyi birbirinden ayırt edebilmek için kullanılan destek vektör makinesi yöntemi, en uygun fonksiyonun tahmin edilmesi temeline dayanır [81]. Sınıflandırmanın temelindeki ana elemanlar, eğitme örnekleri arasında ve her iki sınıfın uç noktasında seçilen destek vektörleridir. Ayrıca genelleme yeteneğinin maksimum olması düzlemin optimum olmasıyla olacaktır.

### 4.2.1. Doğrusal ayrılabilen veriler için DVM

Sınıflandırma destek vektör makineleriyle birlikte genelde {-1, +1} olarak sınıf etiketleri ile gösterilen iki sınıfa ait örneklerin, eğitim verisiyle elde edilen bir karar fonksiyonu yardımıyla birbirinden ayrılması amaçlanır. Karar fonksiyonu kullanılarak eğitim kümesindeki veriyi en uygun şekilde ayırabilecek hiper düzlem bulunmaktadır.

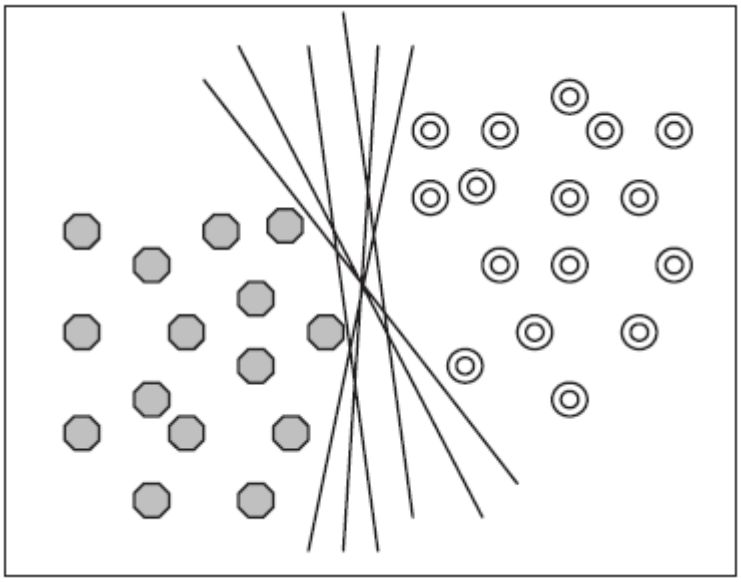
Şekil 4.2'de gösterildiği gibi iki sınıflı verileri birbirinden ayırabilecek birçok hiper düzlem çizilebilir. Fakat DVM'nin amacı kendisine en yakın noktalar arasındaki uzaklığı maksimum seviyeye çıkaran hiper düzlemi bulabilmektir. Şekil 4.3'te ise sınırı maksimuma çıkararak en uygun ayrımı yapan hiper düzleme optimum hiper düzlem, sınır genişliğini sınırlandıran noktalarada destek vektörleri denilmektedir.

Doğrusal olarak ayrılabilen ve iki sınıfı olan bir sınıflandırma işleminde DVM'nin eğitimi için k sayıda örnekten oluşan eğitim verisinin {xi, yi}, i = 1,.....,k = olduğu kabul edilirse, optimum hiper düzleme ait eşitsizlikler aşağıdaki şekilde olur:

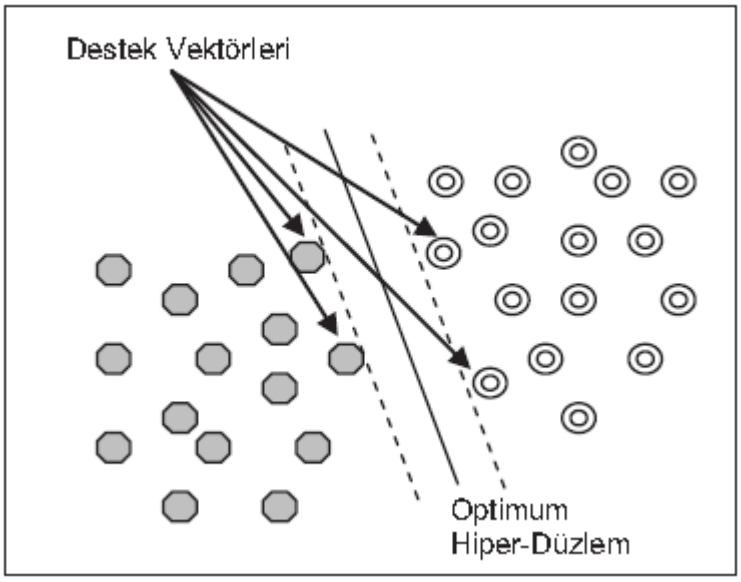
$$w \cdot xi + b \geq +1 \text{ her } y = +1 \text{ için} \quad \textbf{(4.1)}$$

$$w \cdot xi + b \leq +1 \text{ her } y = -1 \text{ için} \quad \textbf{(4.2)}$$

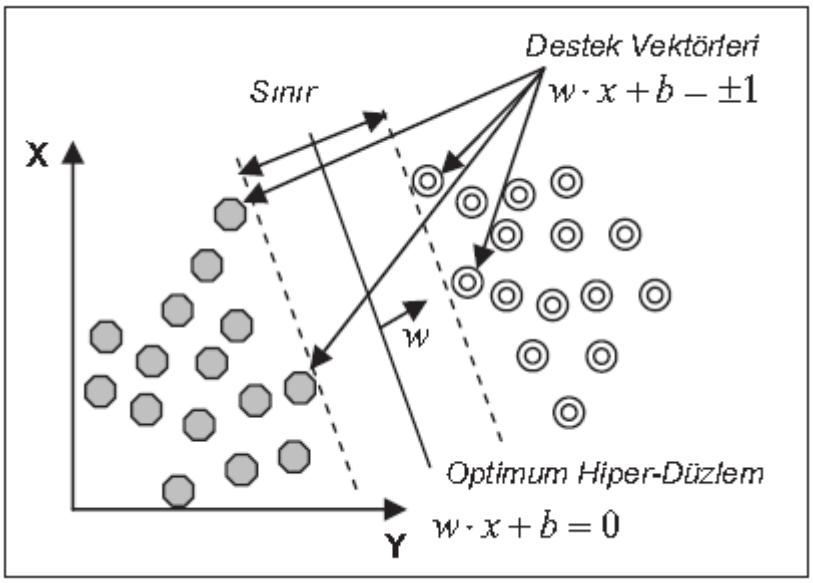
Burada x ∈ RN olup N-boyutlu bir uzayı, y ∈ {-1, +1} sınıf etiketlerini, w ağırlık vektörünü ve "b" de eğilim değerini ifade etmektedir [82]. Optimum hiper düzlemi belirlemeden önce bu düzleme paralel olan ve sınırlarını oluşturacak iki hiper düzlemin belirlenmesi gerekir (Şekil 4.4). Bu hiper düzlemleri oluşturan noktalar destek vektörleri olarak ifade edilir ve bu $w \cdot x_i + b = \pm 1$ şeklinde belirtilmektedir.



**Şekil 4.2:** İki sınıflı problem için hiper düzlemler [94]




**Şekil 4.3:** Destek vektörleri ve optimum hiper düzlem [94]




**Şekil 4.4:** Doğrusal ayrılabilen veri setleri için hiper düzlemin belirlenmesi [94]

Optimum hiper düzlemin sınırını maksimuma çıkarmak için ||w|| ifadesinin minimum hale getirilmesi gerekir. Bu durumda en uygun hiper düzlemi belirlemek için aşağıdaki sınırlı optimizasyon probleminin çözülmesi gerekmektedir.

$$\min[\frac{1}{2}\ ||w||^2] \tag{4.3}$$

Buna bağlı sınırlamalar ise;

$$y_i(w * x_i + b) - 1 \geq 0 \ ve \ y_i \in \{1, -1\} \tag{4.4}$$

olarak ifade edilmektedir [83]. Bu optimizasyon problemi Lagrange denklemleri kullanılarak çözülebilir. İşlem sonrasında;

$$L(w, b, a) = \frac{1}{2}||w||^2 - \sum_{i=1}^{k} a_i y_i (w * x_i + b) + \sum_{i=1}^{k} a_i \tag{4.5}$$

eşitliği elde edilir. Sonuçta, iki sınıflı ve doğrusal olarak ayrılabilen bir problem için karar fonksiyonu aşağıda belirtilen şekilde yazılabilir [82].

$$f(x) = sign\,(\sum_{i=1}^{k} \curlywedge_i\, y_i(x * x_i) + b) \tag{4.6}$$

### 4.2.2. Doğrusal ayrılamayan veriler için DVM

Sınırın maksimuma ve yanlış sınıflandırma hatalarının minimum seviyeye getirilmesi arasındaki denge pozitif değerler alan ve C ile gösterilen bir düzenleme parametresi (0 < C < ∞) tanımlanmasıyla kontrol edilebilir [84]. Yapay değişken ve düzenleme parametresi kullanılarak doğrusal olarak ayrım yapılamayan veriler için optimizasyon problemi:

$$\min[\frac{||w||^2}{2} + C * \sum_{i=1}^{r} \delta_i \tag{4.7}$$

şeklini alır.

Buna bağlı sınırlamalarda;

$$y_i(w * \varphi(x_i) + b) - 1 \geq 1 - \delta_i \tag{4.8}$$

$$\delta_i \geq 0 \text{ ve I = 1,.....N}$$

şeklinde ifade edilir.

Destek vektör makineleri matematiksel olarak $K\left(x_i, x_j\right) = \varphi(x) * \varphi(x_j)$ şeklinde ifade edilen bir kernel fonksiyonu yardımıyla doğrusal olmayan dönüşümler yapılabilmekte olup bu şekilde verilerin yüksek boyutta doğrusal olarak ayrımına olanak sağlamaktadır. Sonuçta, doğrusal olarak ayrılamayan iki sınıflı bir problemin çözümü ile ilgili karar kuralı, kernel fonksiyonu kullanarak aşağıdaki şekilde yazılabilir [82]:

$$f(x) = sign\,(\sum_{i} a_i y_i \varphi(x) * \varphi(x_i) + b \tag{4.9}$$

## 4.3. Centroid Tabanlı Algoritma

Centroid kavramı bir bilgi grubunun orta değerini gösterir. Bir sınıfa ait bilgiler sınıfın ortasındaki bir nokta ile temsil edilir (centroid değeri). Bu şekilde, bir veri grubunun en iyi gösterimi o veri grubunun merkezi değeridir varsayımına dayanılarak her sınıf ayrı bir kütle merkezi ile gösterilir. Centroid tabanlı metin sınıflandırmasında, alıştırma evresi boyunca, her sınıfın kütle merkezi değeri basitçe hesaplanır. Yeni veri örneğinin doğru sınıfını belirleyebilmek için test süresince her centroidin örnek veriyle benzerliği hesaplanır ve benzerliği en fazla olan sınıfa atanır. Centroid tabanlı metin sınıflandırması, vektör uzayı gösterimi modeline dayalı etkili bir metottur. Her belge bir vektör d ile gösterilir ve vektörün her ögesi toplanmış belgelerde bir terime karşılık gelir. Terimlerin büyüklüğünün belirlenmesinde genellikle tf-idf ölçüsü kullanılır. Centroid vektöründe bir öge, bu sınıftaki tüm belgelerin içinde ortalama bir değer belirten bir terime karşılık gelir ve tüm sınıf için bu terim açısından temsili bir değer olarak kabul edilir.

Eğitme verilerinden centroid vektörleri oluşturmada başlıca iki metot vardır [85]. Birincisi, sınıfa ait belge vektörlerinde, centroidlerdeki ögelerin eşleşen terimlerin büyüklüğünün ortasındaki değerin alındığı, aritmetik ortalama centroiddir (AOC). Ci sınıfının centroid vektörü ci şu şekilde oluşur:

$$c_i = \frac{1}{|D_{c_i}|} \sum_{d \in D_{c_i}} d \qquad \textbf{(4.10)}$$

İkinci yöntem terimlerin ortalamalarının yerine, büyüklüklerinin toplamının kullanıldığı kümülüs geometrik centroiddir (KGC).

$$c = \sum_{d \in D_{c_i}} d \qquad \textbf{(4.11)}$$

Basit AOC ve KGC metotlarının farklı çeşitleri vardır. Bunlardan etkili olan çeşit, centroid vektörlerin büyüklüğünü hesaplamada terimler için belge ve sınıf oranının göz önüne alındığı sınıf özelliği centroidtir [86]. Ci sınıfındaki tj teriminin büyüklüğü, cij, şu şekilde hesaplanır:

$$c_{ij} = b^{\frac{|D_{ci},t_j|}{|D_{ci}|}} \log(\frac{|C|}{\left|C_{t_j}\right|} \quad \textbf{(4.12)}$$

b birden büyük bir sabit sayıdır. Denklemin birinci kısmı iç sınıf terimlerin İndeksini gösterirken, ikinci kısmı sınıflar arası terimlerin İndeksini gösterir. Her sınıf bir centroid ile gösterildiğinden, test grubundaki bir belge, benzerlik ölçüsüne göre kategorize edilir. Test belgesi her centroidle kıyaslanır ve en fazla benzerlik seviyesine sahip olan sınıfa atanır. Genel olarak kullanılan benzerlik ölçüsü kosinüs benzerliğidir. Verilen d belgesinde, her sınıfın d belgesi ile benzerliği hesaplanmış ve maksimuma çıkarılmış benzerlik değeri seçilmiştir.

$$sim\ (d, c_i) = \frac{d * c_i}{|d| * |c_i|}, \quad \textbf{(4.13)}$$

$$argmax_i sim(d, c_i) \quad \textbf{(4.14)}$$

## 4.4. Terim Ağırlıklandırma Yöntemleri

Belge vektörlerini vektor uzayı modelinde terimleri kullanarak işlem yaparken, terimlerin sadece belgede bulunup bulunmamasına bakmak yeterli olmayabilir. Bundan dolayı çeşitli terim ağırlıklandırma yöntemleri bulunmuş olup burada TF ve IDF olmak üzere iki farklı bileşen bulunmaktadır.

### 4.4.1. TF (Term Frequency – Terim Sıklığı)

Terim ağırlıklandırmadaki ilk bileşeni terim sıklığı oluşturmaktadır (Term Frequency - TF). Bir belge içerisinde diğer terimlere gore daha fazla bulunan bir terimin oneminin daha fazla olması esasına dayanmaktadır [66]. Terim sıklığı aşağıdaki formülle belirlenebilir:

$$TF_{t,d} = \frac{n_{t,d}}{|d|} \quad \textbf{(4.15)}$$

### 4.4.2. TF-IDF

Metinlerin bulunduğu kümede daha az sayıda bulunan bir terimin ayırt edici özelliğini gösteren ölçüt ters doküman sıklığı (inverse document frequency, IDF) olarak adlandırılır.

$$TFIDF(t,d) = TF\ (d,t), IDF(t) \tag{4.16}$$

Bir metinde çok bulunan fakat diğer metinlerde daha az bulunan bir terimin ağırlığının fazla olduğunu TF ve IDF çarpımı göstermektedir.

$$TFIDF\left(t_k, d_j\right) = \#(t_k, d_j).\log_2 \frac{|T_r|}{|T_r(t_k)|} \tag{4.17}$$

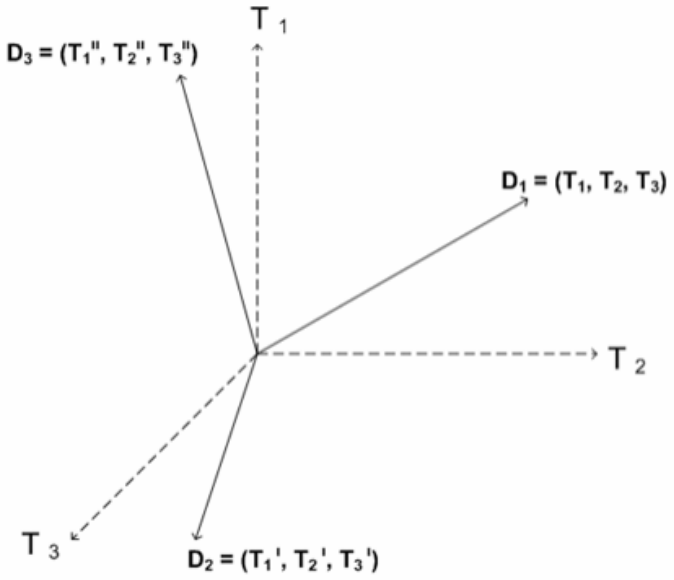
Burada $\#\left(t_k, d_j\right)$, tk kelimesinin dj dokümanı içinde geçme sayısını, $|T_r|$ tüm dokümanları, $|T_r(t_k)|$içinde en az bir kere tk kelimesi geçen dokümanları ifade etmektedir. Bir terimin bir dokümanda bulunma sayısıyla birlikte TF-IDF ağırlığı artmakta, aynı zamanda o terimin tüm doküman uzayında yer alma sayısıyla birlikte azalmaktadır [88]. Metin ağırlıklandırmasında terim dokümanda yoksa dokümanın vektöründe terim sıfır ile ağırlıklandırılmaktadır.

Doküman kümesindeki tüm dokümanlar ve tüm terimler aşağıda Şekil 4.5'teki gibi bir matrisle gösterilebilir. Bu matriste sütunlar dokümanları, satırlar terimleri ifade etmektedir. Şekil incelendiğinde altı doküman ve bu dokümanlarda yer alan on terim gösterilmiştir. İkilik ağırlıklandırmada terim bir dokümanda varsa bir yoksa sıfır olarak ifade edilmekte, terim sıklığı ile ağırlıklandırmada ağırlık terimin dokümanda bulunma sayısı olarak ifade edilmekte ve terim sıklığı – ters doküman sıklığı ile ağırlıklandırmada ağırlık, terim sıklığı ve ters doküman sıklığı değerlerinin çarpımı şeklinde ifade edilmektedir.

| | D1 | D2 | D3 | D4 | D5 | D6 |
|---|---|---|---|---|---|---|
| T1 | | | | | | |
| T2 | | | | | | |
| T3 | | | | | | |
| T4 | | | | | | |
| T5 | | | | | | |
| T6 | | | | | | |
| T7 | | | | | | |
| T8 | | | | | | |
| T9 | | | | | | |
| T10 | | | | | | |
| T11 | | | | | | |

**Şekil 4.5:** Doküman ve terimlerin matris ile gösterimi

Dokümanların terim uzayında ifade edilmesi ise Şekil 4.6'da gösterilmiştir [89]. Bu şekilde terim uzayı üç terimden oluşmuştur. Bu terim uzayında dokümanlara ait vektörler belirtilmektedir.




**Şekil 4.6:** Doküman uzayında vektörlerin gösterimi [89]

## 4.5. Kullanılan Performans Ölçüleri

Performans ölçümünde doğruluk (accuracy), anma (recall), duyarlılık (precision) ve F-ölçütü (F-measure) kullanılmaktadır. Model üzerindeki başarı ölçüldüğü sırada, başarı oranını doğru tespit edebilmek için öğrenme veri kümesini işleme almamak gerekir. Öğrenme veri kümesi üzerinde model oluşturulduktan sonra ilgili model test veri kümesinde sınanır [87].

Bu çalışmada kapsamında karışıklık matrisi kullanılıp doğruluk (accuracy) değeri hesaplanarak başarı oranı belirlenmiştir.

### 4.5.1. Karışıklık matrisi (Confusion matrix)

Bir dokümanın tahmin ve gerçek sınıf değerlerini gösteren matristir. (Çizelge 4.1)

**Çizelge 4.1:** Karışıklık Matrisi

| | | **TAHMİN** | |
|---|---|---|---|
| | | **Negatif** | **Pozitif** |
| **GERÇEK** | **Negatif** | a | b |
| | **Pozitif** | c | d |

### 4.5.2. Doğruluk (Accuracy)

Doğruluk oranı model başarımı ölçümünde kullanılanılan en popüler yöntemlerden biridir. Hesaplama doğru sınıflandırılmış örnek sayısının (TP + TN), toplam örnek sayısına (TP + TN + FP + FN) bölünmesi ile yapılmakta olup aşağıdaki şekilde ifade edilmektedir.

$$Doğruluk = \frac{a + d}{a + b + c + d} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad \textbf{(4.18)}$$

Hata oranı ise yanlış sınıflandırılmış örnek sayısının (FP + FN), toplam örnek sayısına (TP + FP + FN + TN) bölünmesi ile bulunukta olup aşağıdaki şekilde ifade edilmektedir.

$$Hata\ Oranı = \frac{b + c}{a + b + c + d} = \frac{\mathrm{FP}\ +\ \mathrm{FN}}{\mathrm{TP}\ +\ \mathrm{FP}\ +\ \mathrm{FN}\ +\ \mathrm{TN}} \quad \textbf{(4.19)}$$

# 5. DENEYSEL ÇALIŞMALAR

## 5.1. Veri Kümesinin Oluşturulması

Veri toplama aşamasında, twitter4j api ile javada yazılan program ile birlikte Avea, Turkcell ve Vodafone GSM operatörlerinin takipçilerine ait tweetler çekilmiştir. Anahtar kelime olarak şirket isimleri kullanılmış olup toplamda 8379 toplanmıştır. Toplanan bu verileri ait detaylı bilgiler Çizelge 5.1'de gösterilmektedir.

**Çizelge 5.1:** Toplanan veri sayısı

| Veri Seti | Negatif Tweetler | Pozitif Tweetler | Toplam Tweet Sayısı |
|---|---|---|---|
| Avea | 3718 | 559 | 4278 |
| Turkcell | 2157 | 798 | 2956 |
| Vodafone | 702 | 442 | 1145 |

## 5.2. Veri Ön İşleme

Sınıflandırmada kullanılacak verilerin bazen eksik veya tutarsız olduğu görülebilir. Veritabanlarında yer alan eksik veya hatalı veriler gürültü adı verilmektedir. Gürültülü verilerin olması durumunda, bu sorunun giderilmesi beklenmektedir. Aşağıdaki yöntemler bu gibi durumlarda kullanılabilir.

- Gürültülü verilerin veritabanından silinmesi veya yerine yenisinin eklenmesi gerekmektedir.
- Gürültülü verinin yerine sabit bir değer kullanılabilir.
- Tüm verilerin veya bir kısım verilerin ortalaması hesaplanıp gürültülü verilerin yerine bu değer kullanılabilir.
- Gürültülü verilerin yerine, veritabanındaki verilerin tamamı veya belli bir kısmı kullanılarak gürültülü veriler tahmin edilebilir. Elde edilen bu veriler gürültülü verilerin yerine kullanılabilir [90].

Çalışmanın veri işleme aşamasında ilgili tweetlerdeki linkler, kullanıcı adları, noktalama işaretleri, stopwordsler, ve retweetler kaldırılmıştır. Ayrıca aynı cümleler silinmiş olup tüm kelimeler küçük harfe dönüştürülmüştür.

Normalizasyonu tamamlanan veri üzerinde İTÜ Doğal Dil İşleme aracı kullanılarak kelime düzeltme işlemi yapılmıştır. Bu program tüm kelimeleri alt alta yazarak text halinde vermektedir. Yani ilk kelime veri kümesinde bulunan kelime altındaki kelime ise onun düzeltilmiştir halidir. Buradan veri kümesi düzeltme işlemine geçmeden önce bir filtre programı kullanılarak verilerin belirli bir standartta olması sağlanmıştır. Sonrasında, değiştirme programı yardımıyla ile veri kümesindeki hatalı yazılmış kelimeler doğruları ile düzeltilmiştir.

Metinlerde geçen ifadeleri tekileştirip sınıflandırmadaki başarı oranını artırmak amacıyla kelimeler köklerine ayrılmıştır. Hatalı kelimeler üzerinde köklerine ayırma işlemi hatalı olacağından dolayı kök ayırma (stemming) işlemi kelime düzeltme aşamasından sonra zemberek kütüphanesi kullanılarak hazırlanan programla uygulanmıştır.

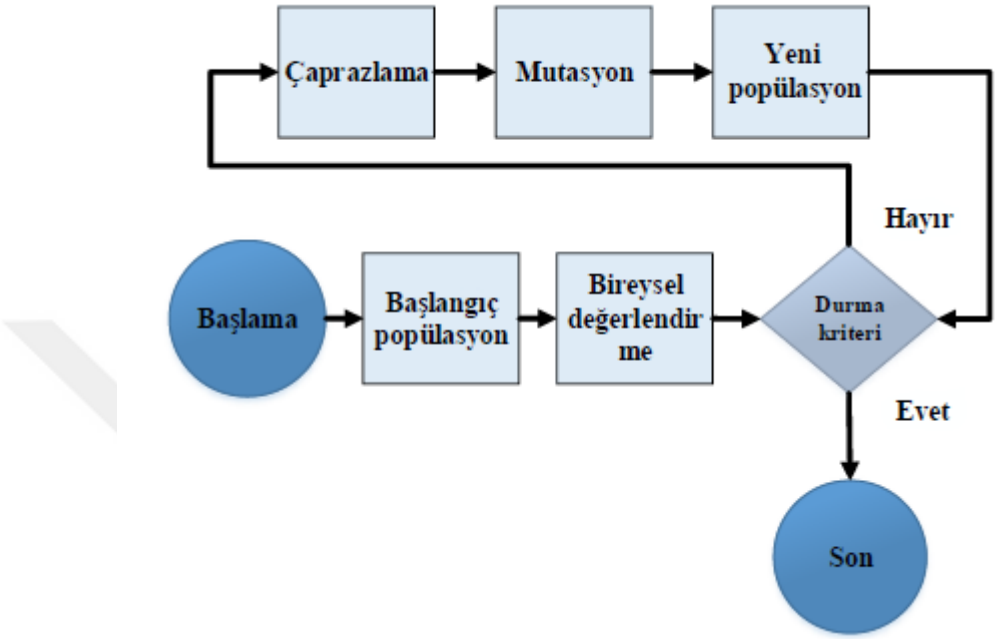
## 5.3. Öznitelik Seçimi

Çalışma kapsamında Gini İndeks, bilgi kazancı düşük hesaplama maliyetleri ve kolay uygulanabilir olmalarından dolayı, genetik algoritma ise boyut indirgemede veriyi sezgisel olarak değerlendirme daha iyi sonuç vereceği düşünülerek kullanılmıştır. Bu algoritmalar yapay sinir ağları, destek vektör makineleri ve centroid tabanlı sınıflandırma algoritmalarına ayrı ayrı entegre edilmiştir.

Bilgi kazancında entropi hesaplaması yapılarak 3 ayrı veri setine göre 200 öznitelik belirlenmiştir. Gini İndeks'te ise yine 200 öznitelik 2 farklı sınıf etiteki bazında hesaplanıp elde edilmiştir.

GA, ise bu çalışma kapsamında önemli bir yere sahiptir. Aşağıda Şekil 5.1'de gösterilen öznitelik seçimi sürecinde üretilen başlangıç popülasyonunu girdi olarak alan GA, popülasyonun her bireyini (kromozom) uygunluk fonksiyonu aracılığıyla değerlendirmektedir. Burada durma kriteri yani iterasyon sayısı kontrol edilir. Çaprazlama ve mutasyon işlemleri GA sonlanana kadar seçilen bireyler üzerinde

yapılır. Bu operatörler yeni bir popülasyon oluşturarak tekrardan değerlendirme aşamasına döner ve durma kriterine erişine kadar işlemler devam eder. Durma kriterini sağlandığında, GA, en iyi sınıflandırma doğruluğuna ve en uygun veya en uyguna yakın bir öznitelik alt kümesi elde eder.




**Şekil 5.1:** GA ile öznitelik seçim süreci [91]

Veri madenciliği sınıflandırma çalışmaları kapsamında yapılan tüm deneylerde verinin %75'i eğitim %25'i ise test kümesine ayrılmıştır. Sütünları özniteliklerden oluşan TF ve TF-IDF matrislerinde ise değeri en yüksek özniteliklerden 200 adet öznitelik üzerinde deneysel çalışmalar yapılmıştır. Yapay sinir ağlarında ise 40 iterasyon ve 20 gizli katman kullanılarak sonuçlar elde edilmiştir.

3 farklı sınıflandırma algoritması, 2 farklı öznitelik seçme ve 1 öznitelik indirgeme algoritmasının kullanıldığı çalışmada Destek Vektör Makineleri (DVM), genetik algoritma, TF ve TF-IDF'in ayrı ayrı kullanıldığı hibrit yöntemin en iyi sonucu verdiği aşağıda Çizelge 5.3 ve Çizelge 5.5'te görülmektedir. Çizelge 5.2'de sadece TF ve DVM sınıflandırma algoritmasının kullanıldığı deneyde Avea ve Vodafone veri setlerinde %100 başarı elde edilmiştir. Turkcell'deki başarı oranı ise yine oldukça yüksek olup %99.5'tir. Çizelge 5.4'te TF-IDF'in sadece 3 sınıflandırma algoritması ile beraber kullanılmasının DVM'deki başarıyı düşürdüğü görülmüştür. Bunun yanı sıra TF-IDF uygulanan diğer algoritmalar aşağıda Çizelge 5.2 ile karşılaştırıldığında

Avea verisi üzerinde yapay sinir ağları uygulaması haricinde diğer deneylerde daha başarılı sonuçlar elde edilmiştir.

**Çizelge 5.2:** TF ile 3 sınıflandırma algoritmasının doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM | Yapay Sinir Ağları | Centroid Tabanlı Alg. |
|---|---|---|---|---|---|
| Avea | Unigram | TF | % 100 | % 87.0 | % 61.9 |
| Turkcell | Unigram | TF | % 99.5 | % 74.2 | % 63.1 |
| Vodafone | Unigram | TF | % 100 | % 74.8 | % 74.5 |

**Çizelge 5.3:** TF ile 3 sınıflandırma alg. + genetik algoritma doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM + Genetik Algoritma | Yapay Sinir Ağları + Genetik Algoritma | Centroid Tabanlı Alg.+ Genetik Algoritma |
|---|---|---|---|---|---|
| Avea | Unigram | TF | **% 100** | % 86.9 | % 86.9 |
| Turkcell | Unigram | TF | **% 100** | % 73.4 | % 74.2 |
| Vodafone | Unigram | TF | **% 100** | % 76.4 | % 74.5 |

**Çizelge 5.4:** TF-IDF ile 3 sınıflandırma algoritmasının doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM | Yapay Sinir Ağları | Centroid Tabanlı Algoritma |
|---|---|---|---|---|---|
| Avea | Unigram | TF-IDF | % 99.8 | % 86.5 | % 69.7 |
| Turkcell | Unigram | TF-IDF | % 99.7 | % 75.6 | % 69.6 |
| Vodafone | Unigram | TF-IDF | % 99.7 | % 75.5 | % 76.9 |

**Çizelge 5.5:** TF-IDF ile 3 sınıflandırma alg. + genetik alg. doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM + Genetik Algoritma | Yapay Sinir Ağları + Genetik Algoritma | Centroid Tabanlı Alg. + Genetik Algoritma |
|---|---|---|---|---|---|
| Avea | Unigram | TF-IDF | **% 100** | % 86.9 | % 87.4 |
| Turkcell | Unigram | TF-IDF | **% 100** | % 73.3 | % 73.5 |
| Vodafone | Unigram | TF-IDF | **% 100** | % 76.4 | % 75.5 |

Aşağıdaki Çizelge 5.6, 5.7, 5.8 ve 5.9'daki deney sonuçları incelendiğinde DVM üzerinde Gini İndeks ve bilgi kazancı algoritmalarının uygulanması başarı oranını artırmamıştır. Bunun yerine deterministik olmayan bir yöntem olarak genetik

algoritma kullanılarak öznitelik boyut indirgemesi yapılarak her 3 veri setinde de %100 başarı elde edilmiştir.

Genel olarak Gini İndeks ve Bilgi Kazancı öznitelik seçimi algoritmalarını karşılaştırdığımızda TF-IDF tekniğinde centroid based algoritmasının TF'e göre daha iyi sonuç verdiğini görmekteyiz (Çizelge 5.8, Çizelge 5.9).

**Çizelge 5.6:** TF ile 3 sınıflandırma algoritması + Gini İndeks doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM + Gini İndeks Alg. | Yapay Sinir Ağları + Gini İndeks Alg. | Centroid Tabanlı Alg. + Gini İndeks Alg. |
|---|---|---|---|---|---|
| Avea | Unigram | TF | % 100 | % 87.9 | % 62.6 |
| Turkcell | Unigram | TF | % 98.9 | % 76.0 | % 66.7 |
| Vodafone | Unigram | TF | % 96.5 | % 80.8 | % 79.4 |

**Çizelge 5.7:** TF ile 3 sınıflandırma algoritması + Bilgi Kazancı doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM + Bilgi Kazancı Alg. | Yapay Sinir Ağları + Bilgi Kazancı Alg. | Centroid Tabanlı Alg. + Bilgi Kazancı Alg. |
|---|---|---|---|---|---|
| Avea | Unigram | TF | % 100.0 | % 87.7 | % 62.3 |
| Turkcell | Unigram | TF | % 99.5 | % 77.3 | % 66.4 |
| Vodafone | Unigram | TF | % 100.0 | % 77.3 | % 80.4 |

**Çizelge 5.8:** TF-IDF ile 3 sınıflandırma alg. + Gini İndeks alg. doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM + Gini İndeks Alg. | Yapay Sinir Ağları + Gini İndeks Alg. | Centroid Tabanlı Alg. + Gini İndeks Alg. |
|---|---|---|---|---|---|
| Avea | Unigram | TF-IDF | % 100 | % 87.2 | % 77.3 |
| Turkcell | Unigram | TF-IDF | % 99.1 | % 75.8 | % 77.3 |
| Vodafone | Unigram | TF-IDF | % 97.2 | % 78.7 | % 81.1 |

**Çizelge 5.9:** TF-IDF ile 3 sınıflandırma alg. + Bilgi Kazancı doğruluk değerleri

| Veri Seti | N-gram | Sıralama Tekniği | DVM + Bilgi Kazancı Alg. | Yapay Sinir Ağları + Bilgi Kazancı Alg. | Centroid Tabanlı Alg. + Bilgi Kazancı Alg. |
|---|---|---|---|---|---|
| Avea | Unigram | TF-IDF | % 100 | % 87.6 | 75.8 |
| Turkcell | Unigram | TF-IDF | % 99.5 | % 75.2 | 75.5 |
| Vodafone | Unigram | TF-IDF | %100 | % 76.6 | 80.1 |

Görüldüğü gibi günlük konuşma dili ile yazılan sosyal medya (twitter) verileri ile önerilen hibrit yöntemle yüksek başarım elde edilebilmektedir. Bu başarının elde edilmesinde sınıflandırma öncesinde uygulanan yöntemlerinde payı oldukça büyüktür. Toplanan ham veri ilk etapta gereksiz ifadelerden ve aynı cümlelerden arındırılıp üzerinde kelime düzeltme (spell correction) ve köklerine ayırma uygulanıp oldukça kaliteli bir hale getirilmiştir. Özellikle kelime düzeltme ve köklerine ayırma işlemleri ile biribirlerine benzer olan kelimeler yakalanmış ve öğrenme kolaylaştırılmıştır. SVM'in yüksek başarısı ile beraber deterministik olmayan genetik algoritma ile en iyi öznitelikler yakalanarak 3 veri setinde de en yüksek başarı elde edilmiştir.

# 6. SONUÇ VE ÖNERİLER

Sınıflandırma başarısını arttırmaya yönelik çalışmalar araştırmacılar tarafından sıkça yapılmaktadır. Etkili bir sınıflandırmada algoritmaların başarısı oldukça önemlidir. Başarıyı etkileyen bir diğer faktör ise veri kümelerinin sahip olduğu niteliklerdir. Gürültülü veya ilgisiz nitelikler sınıflandırmanın başarısını olumsuz yönde etkilemektedir.

Etkili bir sınıflandırma yapmak için veri kümesini en iyi tanımlayan özniteliklerin bulunması veya ilgisiz özniteliklerin atılması çok önemlidir. Çalışma kapsamında kaliteli hale getirilen veriler üzerinde 3 farklı sınıflandırma algoritması (DVM, Yapay Sinir Ağları ve Centroid Tabanlı Algoritma) öznitelik seçme yöntemleri ile beraber uygulanmıştır. Yapılan deneylerde genel anlamda Gini İndeks ve Bilgi Kazancı algoritmaları pek olumlu sonuç vermezken sezgisel bir algoritma olan GA'nın ve ayrıca TF, TF-IDF'in uygulanması ile beraber 3 farklı data üzerinde en yüksek başarı DVM ile elde edilmiştir. Yapılan bu çalışmada, etkin bir hibrit öznitelik seçme modeli önerilmiş ve bu hibrit metotta, GA, yüksek doğruluğa ve küçük boyuta sahip olan en uygun öznitelik alt kümesini bulmaya çalışmıştır. GSM operatörlerinin takipçilerinin atmış oldukları tweetlerin değerlendirildiği bu çalışmada, önerilen hibrit yöntemin, günlük konuşma dili ile yazılan metinler üzerinde yapılacak duygu analizinde başarılı sonuçlar almak için kullanılabileceği gösterilmiştir.

Bu çalışmanın devamı olarak ileride 3 farklı sınıf etiketi kullanılarak farklı n-gramlar ve farklı sınıflandırma algoritmayla deneyler yapılacaktır. Özellikle sınıf etiketi sayısının artmasının başarıyı ne derecede etkileyeceği konusu üzerinde yoğunlaşılacak ve en yüksek başarının elde edileceği bir yöntem önerilecektir.

## KAYNAKLAR


[1] **Szomszor, M., Kostkova, P., & De Quincey, E.** (2010). # Swineflu: Twitter predicts swine flu outbreak in 2009. *In International Conference on Electronic Healthcare*, Springer Berlin Heidelberg sayfa. 18-26.

[2] **Bian, J., Topaloglu, U., & Yu, F.** (2012). Towards large-scale twitter mining for drug-related adverse events. *In Proceedings of the 2012 international workshop on Smart health and wellbeing*, ACM, sayfa. 25-32

[3] **Nguyen, L. T., Wu, P., Chan, W., Peng, W., & Zhang, Y.** (2012) Predicting collective sentiment dynamics from time-series social media. *InProceedings of the first international workshop on issues of sentiment discovery and opinion mining*, ACM, sayfa.6

[4] **Claster, W. B., Dinh, H., & Cooper, M.** (2010). Naïve Bayes and unsupervised artificial neural nets for Cancun tourism social media data analysis. *In Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on* IEEE, sayfa. 158-163

[5] **Bing L.** (2012). "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, cilt 5, sayı. 1, sayfa. 1-167.

[6] **Nasukawa T. And Yi J.** (2003). "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2nd international conference on Knowledge capture*, Sanibel Island, FL, USA.

[7] **Dave K., Lawrence S., and David M. P.** (2003). "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide Web*, ACM.

[8**] Elliott C. D.** (1992). "The Affective Reasoner: A process model of emotions in a multi-agent system," *Northwestern University*, Evanston, IL, USA, 1992.

[9] **Ortony A.** (1990). "The cognitive structure of emotions*", Cambridge university press*, 1990.

[10] **Stevenson R. A., Mikels J. A. and Jam T. W.** (2007). "Characterization of the affective norms for English words by discrete emotional categories," *Behavior Research Methods*, cilt. 39, sayı. 4, sayfa. 1020-1024.

[11] **Eroğul, U.** (2009). *Sentiment analysis in Turkish*, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara

[12] **Taner, B.** (2011). *Feature-Based Sentiment Analysis with Ontologies*, Yüksek Lisans Tezi, Sabancı Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

[13] **Albayrak, N.B.** (2011). *Opinion and Sentiment Analysis Using Natural Language Processing Techniques*, Yüksek Lisans Tezi, Fatih Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

[14**] Akbaş, E.** (2012). *Aspect Based Opinion Mining on Turkish Tweets*, Yüksek Lisans Tezi, Bilkent Üniversitesi, Fen Bilimleri Enstitüsü, Ankara

[15] **Boynukalın, Z. ve Karagöz, P.** (2013). "Emotion analysis on Turkish texts", *Information Sciences and Systems*, sayfa. 159-168.

[16] **Nizam, H. ve Akın, S.S.** (2014). "Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Dengeli ve Dengesiz Veri Setlerinin Performanslarının Karşılaştırılması", *XIX. Türkiye'de İnternet Konferansı*, İzmir.

[17] **Meral, M. ve Diri, B.** (2014). Twitter Üzerinde Duygu Analizi, *IEEE 22. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Trabzon, Trabzon.
[18] **Simsek, M. ve Ozdemir, S.** (2012). Analysis of the relation between Turkish twitter messages and stock market index, Application of Information and Communication Technologies (AICT), *2012 6th International Conference on*, *IEEE*, sayfa.1–4.
[19] **Akba F., Uçan, A., Akçapınar Sezer, E. ve Sever, H.** (2014). "Assessment of Feature Selection Metrics for Sentiment Analyses: Turkish Movie Reviews", *In Proceedings of the 8th European Conference on Data Mining*, Lisbon, Portugal, sayfa.180-184.
[20] **Sevindi, B.İ.** (2013). *Türkçe Metinlerde Denetimli Ve Sözlük Tabanlı Duygu Analizi Yaklaşımlarının Karşılaştırılması*, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
[21] **Özsert, C. M. ve Özgür, A.** (2013). "Word Polarity Detection Using A Multilingual Approach", *In Computational Linguistics and Intelligent Text Processing*, sayfa.75-82.
[22] **Vural, A.G.** (2013). *Sentiment-Focused Web Crawling*, Doktora Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
[23] **Gibbs, R.W.** (1986). On the psycholinguistics of sarcasm., *Journal of Experimental Psychology: General*, 115(1), sayfa.3.
[24] **González-Ibánez, R., Muresan, S. ve Wacholder, N.** (2011). Identifying sarcasm in Twitter: a closer look, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, Volume 2, Association for Computational Linguistics, sayfa.581–586.
[25] **Davidov, D., Tsur, O. ve Rappoport, A**. (2010). Semi-supervised recognition of sarcastic sentences in twitter and amazon, *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, *Association for Computational Linguistics*, sayfa.107–116.
[26] **Pennebaker, J.W., Mehl, M.R. ve Niederhoffer, K.G.** (2003). Psychological aspects of natural language use: Our words, our selves, *Annual review of psychology*, 54(1), sayfa. 547–577.
[27] **Pang, B., Lee, L. ve Vaithyanathan, S.** (2002). Thumbs up?: sentiment classification using machine learning techniques, *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, Volume 10, Association for Computational Linguistics, sayfa.79–86.
[28] **Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y. ve Potts, C.** (2013). Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Citeseer, sayfa.1631–1642.
[29] **Bengio, Y., Schwenk, H., Senécal, J.S., Morin, F. ve Gauvain, J.L.,** (2006). Neural probabilistic language models, Innovations in Machine Learning, *Springer*, sayfa.137–186.
[30] **Jiang, L., Yu, M., Zhou, M., Liu, X. ve Zhao, T.** (2011). Target-dependent twitter sentiment classification, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Cilt. 1, sayfa.151–160.
[31] **Turney, P.D.** (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, *Proceedings of the 40th annual meeting on*

*association for computational linguistics, Association for Computational Linguistics,* sayfa.417–424.
[32] **Pang, B. ve Lee, L.** (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, *Proceedings of the 42nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics*, sayfa.271.
[33] **Nguyen, L.T., Wu, P., Chan, W., Peng, W. ve Zhang, Y.** (2012). Predicting collective sentiment dynamics from time-series social media, *Proceedings of the first international workshop on issues of sentiment discovery and opinion mining*, ACM, sayfa. 6.
[34] **Piramuthu, S.** (1998). Evaluating Feature Selection Methods for Learning in Data Mining Applications. *Proc. 31st Ann. Hawaii Int'l Conf. System Science*, sayfa. 294-301.
[35] **Martin-Bautista, M. J., Vila, M. A.** (1999). A Surandy of Genetic Feature Selection in Mining Issues. *Proc. 1999 Congress on Evolutionary Computation* (CEC '99), sayfa. 1314-1321.
[36] **Messer, K., Kittler, J**. (1997). Using feature selection to aid an iconic search through an image database. *Proc. EEE Int'l Conf. Acoustics, Speech, and Signal processing (ICASSP)*, sayı. 4, sayfa. 2605-2608.
[37**] Liu, Y., Dellaert, F.** (1998). A classification based similarity metric for 3D image retrieval. *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, sayfa. 800-805.
[38] **Puuronen, S., Tsymbal, A. and Skrypnik, I.** (2000). Advanced local feature selection in medical diagnostics. *Proc. 13th IEEE Symp. Computer-Based Medical Systems*, sayfa. 25-30.
[39] **Holland J.** (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. *University of Michigan Press*, Ann Arbor, MIT Press, Cambridge.
[40**] Siedlecki, W., Sklansky, J.** (1989). A note on genetic algorithms for Large-Scale feature selection. *Pattern Recognition Letters*, sayı. 10, sayfa. 335-347.
[41] **Brill, F. Z., Brown, D. E. and Martin, W. N.** (1992). Fast genetic selection of features for neural network classifiers. *IEEE Trans. Neural Networks*, 3(2), 324-328.
[42] **Raymer, M. L. Punch, W.F., Goodman, E. D., Kuhn L. A. and Jain A. K.** (2000). Dimensionality reduction using genetic algorithms. *IEEE Trans. Evolutionary Computation*, 4(2), sayfa.164-171.
[43] **Jog, P., Suh, J. and Gucht, D.** (1989). The Effect of population size, heuristic crossoandr and local improandment on a genetic algorithm for the traandling Salesman problem. *Proc. Int'l Conf. Genetic Algorithms*, sayfa.110-115.
[44] **Zheng, X., Julstrom, B.A. and Cheng, W.** (1997). Design of andctor quantization codebooks using a genetic algorithm. *Proc. IEEE Int'l Conf. Evolutionary Computation*, sayfa. 525-529.
[45] **Saeys, Y., Degroeve, S., Aeyels, D., Rouzé, P., & Van de Peer, Y.** (2004). Feature selection for splice site prediction: a new method using EDA-based feature ranking. *BMC bioinformatics*, 5(1), sayfa. 64.
[46] **Tan, P., Steinbach, M. and Kumar, V.** (2005). Introduction to Data Mining. *Addison Wesley*, 1st edition.
[47] **Molina, L., Belanche, L. and Nebot, A.** (2002). Feature selection algorithms: A surandy and experimental evaluation. *In Proceedings of the 2002 IEEE International Conference on Data Mining* (ICDM'02), sayfa. 306–313.

[48] **Guyon, I., Elisseef, A.** (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, sayı. 3, sayfa. 1157–1182.
[49] **Dash, M., Liu H.** ( 1997). Feature Selection for Classification. *Intelligent Data Analysis*, 1(3), sayfa.131-156.
[50] **Bhanu, B., Dudgeon, D., Zelnio, E., Rosenfeld, A., Casaseut, D. and Reed, I. (Eds).** (1997). Special issue on automatic target recognition, *IEEE Transactions on Image Processing*, 6(1).
[51] **Bhanu, B., Poggio, T. (Eds)** (1994). Special section on machine learning in computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9).
[52] **Punch, W., Goodman, E**. (1993). Further research on feature selection and classification using genetic algorithms, *Proceedings of the Fifth International Conference on Genetic Algorithms*, sayfa. 557–564.
[53] **Matsui, K., Suganami, Y. and Kosugi, Y.** (1999). Feature selection by genetic algorithm for MRI segmentation. *Systems and Computers in Japan*, 30 (7), sayfa. 69–78.
[54] **DELİBAŞ, A.** (2008). *Doğal Dil İşleme ile Türkçe Yazım Hatalarının Denetlenmesi*, İstanbul Teknik Üniversitesi FBE, Yüksek Lisans Tezi.
[55] **ÖZBİLİCİ, A.** (2006). *Türkçe Doğal Dili Anlamada İlişkisel Ayrık Bilgiler Modeli ve Uygulaması*, Sakarya Üniversitesi FBE, Yüksek Lisans Tezi
[56] **NABİYEV, V.V.** (2010). Yapay Zeka: İnsan-Bilgisayar Etkileşimi, *Seçkin Yayıncılık*, 3. Baskı, Ankara.
[57] **KESGİN, F.** (2007). "**Türkçe Metinler için Konu Belirleme Sistemi**", İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.
[58] **SAY, B.** (2003). *Türkçe İçin Biçimbirimsel ve Sözdizimsel Olarak İşaretlenmiş Ağaç Yapılı Bir Derlem Oluşturma*, TÜBİTAK EEEAG Projesi.
[59] **Thelwall, M., Buckley, K., Paltoglou, G., Cai, D. ve Kappas, A.** (2010). Sentiment strength detection in short informal text, *Journal of the American Society for Information Science and Technology*, 61(12), sayfa. 2544–2558.
[60] **Ortigosa, A., Martin, J.M. ve Carro, M. R.** (2013) Sentiment analysis in Facebook and its application to e-learning. *Computers in Human Behavior*, sayfa.1-15.
[61] **Sevindi, B.İ.** (2013). *Türkçe metinlerde denetimli ve sözlük tabanlı duygu analizi yaklaşımlarının karşılaştırılması*, Gazi Üniversitesi Fen Bilimleri Enstitüsü Ankara, sayfa.1
[62] **İskender, E.** (2016). *Sosyal Medya Mesajlarında Müşteri Memnuniyetinin Fuzzy Sentiment Analizi İle Ölçülmesi*, Marmara Üniversitesi Sosyal Bilimler Enstitüsü, sayfa. 6-7
[63] **Liu, H.** (2010). Feature selection. *In Encyclopedia of Machine Learning*, sayfa. 402–406.
[64] **Lal, T., Chapelle, O., Weston, J. and Elisseeff, A.** (2006). Embedded methods. In Isabelle Guyon, Masoud Nikraandsh, Steand Gunn, and Lotfi Zadeh, editors, *Feature Extraction*, sayfa. 137–165.
[65] **Guyon, I., Elisseef, A**. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, sayı. 3, sayfa. 1157–1182.
[66] **Manning, C.D., Raghavan, P. ve Schutze, H.** (2009). An Introduction to Information Retrieval, *Cambridge University Press*, Cambridge, England.

[67] **Naqvi, G.** (2012). *A hybrid filter-wrapper approach for feature selection*. International Master's Thesis, Studies from the Department of Technology at Örebro Uniandrsity, sayfa. 0-104.
[68] **Korde, V., & Mahender, C. N.** (2012). Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, 3(2), sayfa. 85.
[69] **Alibeigi, M., Hashemi, S., & Hamzeh, A.** (2009). Unsupervised feature selection using feature density functions. *International Journal of Electrical and Electronics Engineering*, 3(7), sayfa. 394-399.
[70] **Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z.** (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, 33(1), sayfa. 1-5.
[71] **Pohl, I.** (1970). Bi-directional Search. IBM T.J. Watson Research Center, 1970-Database searching, sayfa. 27.
[72] <http://www.webcitation.org/6VERCiLDF>, alındığı tarih: 10.06.2016
[73] **Sever, H., Oğuz, B.** (2002). Veritabanlarında bilgi keşfine formel bir yaklaşım: kısım I: Eşleştirme sorguları ve algoritmalar. *Bilgi Dünyası*, 3(2), sayfa. 173-204.
[74] **Alan, M. A.** (2012). Veri madenciliği ve lisansüstü öğrenci verileri üzerine bir uygulama. *Dumlupınar Üniversitesi Sosyal Bilimler Dergisi*, (33).
[75] **Özekes, S.** (2003). Veri madenciliği modelleri ve uygulama alanları. *İstanbul Commerce University Journal of Science*, 3(3), sayfa. 65-82.
[76] **Öztemel, E.** (2003). Yapay Sinir Ağları*, Papatya Yayıncılık*, İstanbul
[77] **Aktaş, M., Okumuş, H. İ.** (2003). Doğrudan Moment Kontrollü Asenkron Motorun Stator Direncinin Yapay Sinir Ağı ile Kestirimi, *International XII. Turkish Symposium on Artificial Intelligence and Neural Networks*
[78] **Elmas Ç.** (2003). Yapay Sinir Ağları, *Seçkin Yayıncılık*, Ankara.
[79] **Elmas, M.** (2012). *Destek Vektör Makineleri ile Fiyat Tahminleri ve Kuyumculuk Sektöründe Bir Uygulama*, Yüksek Lisans Tezi İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
[80] **ÖZKAN, Y.** (2008). Veri Madenciliği Yöntemleri, *Papatya Yayınları*, İstanbul,
[81] **WANG, L.** (2005). Support Vector Machines: Theory and Applications, *Springer*, New York, sayfa. 1434-9922.
[82] **Osuna, E.E., Freund, R., Girosi, F. (**1997). Support Vector Machines: Training and Applications, *Massachusetts Institute of Technology and Artificial Intelligence Laboratory*, Massachusetts.
[83] **Vapnik, V.N.** (1995). The Nature of Statistical Learning Theory, *Springer-Verlag*, New York.
[84] **Cortes, C., Vapnik, V.** (1995). Support-Vector Network, Machine Learning, 20(3), sayfa. 273–297.
[85] **Guan, H., Zhou, J., & Guo, M.** (2009). A class-feature-centroid classifier for text categorization. *In Proceedings of the 18th international conference on World wide web* ACM, sayfa. 201-210
[86] **Tan, S.** (2008). An improved centroid classifier for text categorization. *Expert Syst. Appl.* 35 (1–2), sayfa. 279–285
[87] **Kırmacı B.** (2015). *Müzik Üst-Veri Tahmini için Türkçe Şarkı Sözü Madenciliği*, Yüksek Lisans Tezi, Başkent Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul
[88] **Cardoso-Cachopo, A., & Oliveira, A. L.** (2006). Empirical evaluation of centroid-based models for single-label text categorization. *INSEC-ID Technical Report*,7, 2006.

[89] **Salton, G., Wong, A., & Yang, C. S.** (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), sayfa. 613-620.
[90] **Özkan, Y.** (2008). Veri madenciliği yöntemleri. *Papatya Yayıncılık Eğitim*
[91] **Moghaddam, S. A. V.** (2014). *Etkin Sınıflandırma İçin Genetik Algoritma Tabanlı Öznitelik Alt Küme Seçimi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü.
[92] **Çınar, D.** (2007). *Hidroelektrik Enerji Üretiminin Hibrid Bir Model ile Tahmini*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
[93] **Deb, K.** (2001). *Multiobjective Optimization Using Evalutionary Algorithms*, John Wiley & Sons, England.
[94] **Kavzoğlu, T., & Çölkesen, İ.** (2010). Destek vektör makineleri ile uydu görüntülerinin sınıflandırılmasında kernel fonksiyonlarının etkilerinin incelenmesi. *Harita Dergisi*, 144(7), sayfa. 73-82.

**EKLER**

**Ek-A:** Destek Vektör Makinesi ve Gini İndeks Algoritması Kullanarak Sınıflandırma
**Ek-B:** Yapay Sinir Ağları ve Gini Index Algoritması Kullanarak Sınıflandırma
**Ek-C:** Centroid Tabanlı Algoritma ve Gini Index Algoritması Kullanarak Sınıflandırma
**Ek-D:** Destek Vektör Makinesi ve Bilgi Kazancı Algoritması Kullanarak Sınıflandırma
**Ek-E:** Yapay Sinir Ağları ve Bilgi Kazancı Algoritması Kullanarak Sınıflandırma
**Ek-F:** Centroid Tabanlı Algoritma ve Bilgi Kazancı Algoritması Kullanarak Sınıflandırma
**Ek-G:** Destek Vektör Makinesi ve Genetik Algoritma Kullanarak Sınıflandırma
**Ek-H:** Yapay Sinir Ağları ve Genetik Algoritma Kullanarak Sınıflandırma
**Ek-I:** Centroid Tabanlı Algoritma ve Genetik Algoritma Kullanarak Sınıflandırma

**Ek-A:**

```
data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:1000);
targets=data(1:end-1,end);
targets(targets==0)=2;

%% öznitelik Seçimi: Gini İndeks
number_of_features = 200;
fsOutput = fsGini(inputs, targets);
oldinputs=inputs';
inputs = inputs(:, fsOutput.fList(1:number_of_features));

inputs = inputs';
targets = targets';

inputs;
targets;


addpath('libsvm-mat-3.0-1');

divideParam.trainRatio= 0.75;
divideParam.valRatio =0.0;
divideParam.testRatio = 0.25;

[divideParam.trainInd,divide.ParamvalInd,divideParam.testInd]=div
ideint(size(inputs,2),divideParam.trainRatio,divideParam.valRatio
,divideParam.testRatio);



train.inputs=inputs(:,divideParam.trainInd);
train.targets=targets(:,divideParam.trainInd);
test.inputs=inputs(:,divideParam.testInd);
test.targets=targets(:,divideParam.testInd);

model = svmtrain(train.targets', train.inputs', '-c 1 -g 1');
[predict_label, accuracy, dec_values] = svmpredict(targets',
inputs', model);

all.outputs=predict_label'
test.outputs=all.outputs(divideParam.testInd)
train.outputs=all.outputs(divideParam.trainInd)

all.outputs2=cevir1(all.outputs,2)';
train.outputs2=all.outputs2(divideParam.trainInd);
test.outputs2=all.outputs2(divideParam.testInd);

all.targets=targets
all.targets2=cevir1(all.targets,2)'
train.targets2=all.targets2(divideParam.trainInd);
test.targets2=all.targets2(divideParam.testInd);
```

```
plotconfusion(train.targets2,train.outputs2,'train',test.targets2
,test.outputs2,'test',all.targets2,all.outputs2,'all');
metrics=metrik(all.targets',all.outputs')
```

**Ek-B:**

```
data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:1000);
targets=data(1:end-1,end);
targets(targets==1)=2;
targets(targets==0)=1;
%targets(targets==-1)=1;

%% öznitelik seçimi: Gini İndeks
number_of_features = 200;
fsOutput = fsGini(inputs, targets);
inputs = inputs(:, fsOutput.fList(1:number_of_features));

inputs = inputs';
targets = targets';
targets=cevir1(targets,2)';

inputs;
targets;


hiddenLayerSize = 20;

net = patternnet(hiddenLayerSize);
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


net.divideFcn = 'divideint';
net.divideMode = 'sample';

net.divideParam.trainRatio= 0.75;
net.divideParam.valRatio =0.0;
net.divideParam.testRatio = 0.25;

net.trainFcn = 'trainlm';  % Levenberg-Marquardt
net.performFcn = 'mse';
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};

net.trainParam.epochs=40;
[net,tr] = train(net,inputs,targets);

outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
testPerformance = perform(net,testTargets,outputs)

view(net)
```

```
plotconfusion(trainTargets,outputs,'train',testTargets,outputs,'t
est',targets,outputs,'all');

targets1=terscevir(targets);
outputs1=outputs>0.5;
outputs1=terscevir(outputs1);
metrics=metrik(targets1',outputs1')


%% Gini İndeks

function [out] = fsGini(X,Y)
    [~,n] = size(X);
    W = zeros(n,1);

    for i=1:n
        values = unique(X(:,i));
        v = size(values,1);
        W(i) = 0.5;
        for j=1:v
            left_Y = Y(X(:,i) <= values(j));
            right_Y = Y(X(:,i) > values(j));

            gini_left = 0;
            gini_right = 0;

            for k=min(Y):max(Y)
               gini_left = gini_left + (size(left_Y(left_Y ==
k),1)/size(left_Y,1))^2;
               gini_right = gini_right + (size(right_Y(right_Y ==
k),1)/size(right_Y,1))^2;
            end
            gini_left = 1-gini_left;
            gini_right = 1-gini_right;

            current_gini = (size(left_Y,1)*gini_left +
size(right_Y,1)*gini_right)/size(Y,1);
            if current_gini<W(i)
                W(i) = current_gini;
            end
        end
    end
    [out.W out.fList]= sort(W);
    out.prf = -1;
end
```

**Ek-C:**

```
data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:1000);
targets=data(1:end-1,end);
targets(targets==0)=2;
number_of_features = 200;
fsOutput = fsGini(inputs, targets);
oldinputs=inputs';
inputs = inputs(:, fsOutput.fList(1:number_of_features));

inputs = inputs';
targets = targets';

inputs;
targets;


divideParam.trainRatio= 0.75;
divideParam.valRatio =0.0;
divideParam.testRatio = 0.25;
nn = size(inputs,2);
ii = randperm(nn);
nt = round(nn*divideParam.trainRatio);
nv = round(nn*divideParam.valRatio);
nte = nn-nt-nv;
divideParam.trainInd = ii(1:nt);
divideParam.valInd = ii(nt+1:nt+nv);
divideParam.testInd = ii(nt+nv+1:nt+nv+nte);



train.inputs=inputs(:,divideParam.trainInd);
train.targets=targets(:,divideParam.trainInd);
test.inputs=inputs(:,divideParam.testInd);
test.targets=targets(:,divideParam.testInd);
c1.ind =(train.targets==1);
c2.ind=(train.targets==2);
c1.inputs=train.inputs(:,c1.ind);
c2.inputs=train.inputs(:,c2.ind);

c1.center=sum(c1.inputs,2).*(1/size(c1.inputs,2));
c2.center=sum(c2.inputs,2).*(1/size(c2.inputs,2));
c1.center=normc(c1.center); % normalize et
c1.center_test=repmat(c1.center,1,size(divideParam.testInd,2));
c2.center=normc(c2.center);
c2.center_test=repmat(c2.center,1,size(divideParam.testInd,2));


test.inputs_normalize=normc(test.inputs);
test.sonuc1=dot(c1.center_test,test.inputs_normalize,1);
test.sonuc2=dot(c2.center_test,test.inputs_normalize,1);
test.outputs=test.sonuc1>test.sonuc2;
test.outputs=double(test.outputs);
test.outputs(test.outputs==0)=2;
test.outputs2=cevir1(test.outputs,2)';
test.targets2=cevir1(test.targets,2)';
```

```
c1.center_train=repmat(c1.center,1,size(divideParam.trainInd,2));
c2.center_train=repmat(c2.center,1,size(divideParam.trainInd,2));

train.inputs_normalize=normc(train.inputs);
train.sonuc1=dot(c1.center_train,train.inputs_normalize,1);
train.sonuc2=dot(c2.center_train,train.inputs_normalize,1);
train.outputs=train.sonuc1>train.sonuc2;
train.outputs=double(train.outputs);
train.outputs(train.outputs==0)=2;
train.outputs2=cevir1(train.outputs,2)';
train.targets2=cevir1(train.targets,2)';

c1.center_all=repmat(c1.center,1,size([divideParam.trainInd
divideParam.testInd],2));
c2.center_all=repmat(c2.center,1,size([divideParam.trainInd
divideParam.testInd],2));

all.inputs_normalize=normc(inputs);
all.sonuc1=dot(c1.center_all,all.inputs_normalize,1);
all.sonuc2=dot(c2.center_all,all.inputs_normalize,1);
all.outputs=all.sonuc1>all.sonuc2;
all.outputs=double(all.outputs);
all.outputs(all.outputs==0)=2;
all.outputs2=cevir1(all.outputs,2)';
all.targets=targets;
all.targets2=cevir1(targets,2)';


plotconfusion(train.targets2,train.outputs2,'train',test.targets2
,test.outputs2,'test',all.targets2,all.outputs2,'all');
metrics=metrik(all.targets',all.outputs')
```

**Ek-D:**

```
clear;
close all;

data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:1000);
targets=data(1:end-1,end);
targets(targets==0)=2;
%targets(targets==0)=1;
%targets(targets==-1)=1;

% öznitelik seçimi: Gini İndeks

javaaddpath('C:\Users\PC-COM\Desktop\Gini-InfoGain-SVM-
Centroid\weka\weka.jar');

number_of_features = 200;
fsOutput = fsInfoGain(inputs, targets);
inputs = inputs(:, fsOutput.fList(1:number_of_features));

inputs = inputs';
targets = targets';
inputs;
targets;


addpath('libsvm-mat-3.0-1');

divideParam.trainRatio= 0.75;
divideParam.valRatio =0.0;
divideParam.testRatio = 0.25;

[divideParam.trainInd,divide.ParamvalInd,divideParam.testInd]=div
ideint(size(inputs,2),divideParam.trainRatio,divideParam.valRatio
,divideParam.testRatio);


train.inputs=inputs(:,divideParam.trainInd);
train.targets=targets(:,divideParam.trainInd);
test.inputs=inputs(:,divideParam.testInd);
test.targets=targets(:,divideParam.testInd);

model = svmtrain(train.targets', train.inputs', '-c 1 -g 1');
[predict_label, accuracy, dec_values] = svmpredict(targets',
inputs', model);

all.outputs=predict_label'
test.outputs=all.outputs(divideParam.testInd)
train.outputs=all.outputs(divideParam.trainInd)

all.outputs2=cevir1(all.outputs,2)';
train.outputs2=all.outputs2(divideParam.trainInd);
test.outputs2=all.outputs2(divideParam.testInd);

all.targets=targets
all.targets2=cevir1(all.targets,2)'
```

```
train.targets2=all.targets2(divideParam.trainInd);
test.targets2=all.targets2(divideParam.testInd);


plotconfusion(train.targets2,train.outputs2,'train',test.targets2
,test.outputs2,'test',all.targets2,all.outputs2,'all');
metrics=metrik(all.targets',all.outputs')
```

**Ek-E:**

```
data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:1000);
targets=data(1:end-1,end);
targets(targets==1)=2;
targets(targets==0)=1;
%targets(targets==-1)=1;

%% öznitelik seçimi: Bilgi kazancı
number_of_features = 200;
fsOutput = fsInfoGain(inputs, targets);
inputs = inputs(:, fsOutput.fList(1:number_of_features));

inputs = inputs';
targets = targets';
targets=cevir1(targets,2)';
inputs;
targets;


hiddenLayerSize = 20;

net = patternnet(hiddenLayerSize);

net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
net.outputs{2}.processFcns = {'removeconstantrows','mapminmax'};


net.divideFcn = 'divideint';
net.divideMode = 'sample';

net.divideParam.trainRatio= 0.75;
net.divideParam.valRatio =0.0;
net.divideParam.testRatio = 0.25;

net.trainFcn = 'trainlm';  % Levenberg-Marquardt
net.performFcn = 'mse';
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
  'plotregression', 'plotfit'};

net.trainParam.epochs=40;
[net,tr] = train(net,inputs,targets);

outputs = net(inputs);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

trainTargets = targets .* tr.trainMask{1};
valTargets = targets  .* tr.valMask{1};
testTargets = targets  .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)
testPerformance = perform(net,testTargets,outputs)
```

```
view(net)

plotconfusion(trainTargets,outputs,'train',testTargets,outputs,'t
est',targets,outputs,'all');

targets1=terscevir(targets);
outputs1=outputs>0.5;
outputs1=terscevir(outputs1);
metrics=metrik(targets1',outputs1')



% Bilgi Kazancı

function [out] =  fsInfoGain(X,Y)

nF = size(X,2);

t = weka.attributeSelection.InfoGainAttributeEval();
t.buildEvaluator(wekaCategoricalData(X, SY2MY(Y)));

out.W = zeros(1,nF);

for i =1:nF;
    out.W(i) = t.evaluateAttribute(i-1);
end


[foo, out.fList] = sort(out.W, 'descend');
out.prf = -1;
end
```

**Ek-F:**

```
data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:1000);
targets=data(1:end-1,end);
targets(targets==0)=2;

javaaddpath('C:\Users\PC-COM\Desktop\Gini-InfoGain-SVM-
Centroid\weka\weka.jar');

% önitelik seçimi : bilgi kazancı
number_of_features = 200;
fsOutput = fsInfoGain(inputs, targets);
inputs = inputs(:, fsOutput.fList(1:number_of_features));

inputs = inputs';
targets = targets';


inputs;
targets;


divideParam.trainRatio= 0.75;
divideParam.valRatio =0.0;
divideParam.testRatio = 0.25;

nn = size(inputs,2);
ii = randperm(nn);
nt = round(nn*divideParam.trainRatio);
nv = round(nn*divideParam.valRatio);
nte = nn-nt-nv;
divideParam.trainInd = ii(1:nt);
divideParam.valInd = ii(nt+1:nt+nv);
divideParam.testInd = ii(nt+nv+1:nt+nv+nte);


train.inputs=inputs(:,divideParam.trainInd);
train.targets=targets(:,divideParam.trainInd);
test.inputs=inputs(:,divideParam.testInd);
test.targets=targets(:,divideParam.testInd);
c1.ind =(train.targets==1);
c2.ind=(train.targets==2);
c1.inputs=train.inputs(:,c1.ind);
c2.inputs=train.inputs(:,c2.ind);

c1.center=sum(c1.inputs,2).*(1/size(c1.inputs,2));
c2.center=sum(c2.inputs,2).*(1/size(c2.inputs,2));
c1.center=normc(c1.center); % normalize et
c1.center_test=repmat(c1.center,1,size(divideParam.testInd,2));
c2.center=normc(c2.center);
c2.center_test=repmat(c2.center,1,size(divideParam.testInd,2));


test.inputs_normalize=normc(test.inputs);
test.sonuc1=dot(c1.center_test,test.inputs_normalize,1);
```

```
test.sonuc2=dot(c2.center_test,test.inputs_normalize,1);
test.outputs=test.sonuc1>test.sonuc2;
test.outputs=double(test.outputs);
test.outputs(test.outputs==0)=2;
test.outputs2=cevir1(test.outputs,2)';
test.targets2=cevir1(test.targets,2)';

c1.center_train=repmat(c1.center,1,size(divideParam.trainInd,2));
c2.center_train=repmat(c2.center,1,size(divideParam.trainInd,2));

train.inputs_normalize=normc(train.inputs);
train.sonuc1=dot(c1.center_train,train.inputs_normalize,1);
train.sonuc2=dot(c2.center_train,train.inputs_normalize,1);
train.outputs=train.sonuc1>train.sonuc2;
train.outputs=double(train.outputs);
train.outputs(train.outputs==0)=2;
train.outputs2=cevir1(train.outputs,2)';
train.targets2=cevir1(train.targets,2)';

c1.center_all=repmat(c1.center,1,size([divideParam.trainInd
divideParam.testInd],2));
c2.center_all=repmat(c2.center,1,size([divideParam.trainInd
divideParam.testInd],2));

all.inputs_normalize=normc(inputs);
all.sonuc1=dot(c1.center_all,all.inputs_normalize,1);
all.sonuc2=dot(c2.center_all,all.inputs_normalize,1);
all.outputs=all.sonuc1>all.sonuc2;
all.outputs=double(all.outputs);
all.outputs(all.outputs==0)=2;
all.outputs2=cevir1(all.outputs,2)';
all.targets=targets;
all.targets2=cevir1(targets,2)';

plotconfusion(train.targets2,train.outputs2,'train',test.targets2
,test.outputs2,'test',all.targets2,all.outputs2,'all');
metrics=metrik(all.targets',all.outputs')
```

**Ek-G:**

```
% Genetik Algoritma Fonksiyonu

global dizi;
global scores;
global netler;
global predict_label;
global divideParam;
dizi={};
scores={};
netler={};

data=csvread('Data.csv',1,1);
inputs=data(1:end-1,1:200)';
targets=data(1:end-1,end)';
targets(targets==1)=2;
targets(targets==0)=1;

options = gaoptimset;
options = gaoptimset(options,'PopulationType', 'bitstring');
options = gaoptimset(options,'PopulationSize', 50);
options = gaoptimset(options,'EliteCount', 2);
options = gaoptimset(options,'CreationFcn', @gacreationuniform);
options = gaoptimset(options,'SelectionFcn',
@selectionstochunif);
options = gaoptimset(options,'CrossoverFcn',
@crossoversinglepoint);
options = gaoptimset(options,'MutationFcn', {  @mutationuniform
[] });

my_output= @(options,state,flag)
myoutput(options,state,flag,fid1);
myfit=@(x)myfuncSVM(x,inputs,targets);
options = gaoptimset(options,'OutputFcns', my_output);
options = gaoptimset(options,'Display', 'iter');
options = gaoptimset(options,'PlotFcns', { @gaplotbestf });
[x,fval,exitflag,output,population,score] = ...
    ga(myfit,4,[],[],[],[],[],[],[],options);
s=int2str(x);
index=find(strncmp(s,dizi,length(s))==1);
score=cell2mat(scores(index))  ;
all.outputs=predict_label'
test.outputs=all.outputs(divideParam.testInd)
train.outputs=all.outputs(divideParam.trainInd)

all.outputs2=cevir1(all.outputs,2)';
train.outputs2=all.outputs2(divideParam.trainInd);
test.outputs2=all.outputs2(divideParam.testInd);

all.targets=targets
all.targets2=cevir1(all.targets,2)'
train.targets2=all.targets2(divideParam.trainInd);
test.targets2=all.targets2(divideParam.testInd);
```

```
plotconfusion(train.targets2,train.outputs2,'train',test.targets2
,test.outputs2,'test',all.targets2,all.outputs2,'all');
metrics=metrik(all.targets',all.outputs')

%SVM Fonksiyonu

    global dizi;
    global scores;
    global netler;
    global predict_label;
    global divideParam;


    s=int2str(x);

    if isempty(find(strncmp(s,dizi,length(s))==1, 1))

        if (all(x==0)==1)
            score=inf;
            dizi(length(dizi)+1)={s};
            scores(length(scores)+1)={score};
            hiddenLayerSize = 20;
            net = patternnet(hiddenLayerSize);
            netler(length(netler)+1)={net};
            return
        end

        mask=x;
        masklog=logical(mask);
        inputs=inputs(masklog,:);



addpath('libsvm-mat-3.0-1');

divideParam.trainRatio= 0.75;
divideParam.valRatio =0.00;
divideParam.testRatio = 0.25;

[divideParam.trainInd,divide.ParamvalInd,divideParam.testInd]=div
ideint(size(inputs,2),divideParam.trainRatio,divideParam.valRatio
,divideParam.testRatio);

train.inputs=inputs(:,divideParam.trainInd);
train.targets=targets(:,divideParam.trainInd);
test.inputs=inputs(:,divideParam.testInd);
test.targets=targets(:,divideParam.testInd);

model = svmtrain(train.targets', train.inputs', '-c 1 -g 1');
[predict_label, accuracy, dec_values] = svmpredict(targets',
inputs', model);


         score=accuracy(2);
        dizi(length(dizi)+1)={s};
        scores(length(scores)+1)={score};
```

```
    else
        index=find(strncmp(s,dizi,length(s))==1);
position index
        score=cell2mat(scores(index))  ;
    end
end
```

**Ek-H:**

```
%Genetik Algoritma Fonksiyonu

global dizi;
global scores;
global netler;
dizi={};
scores={};
netler={};

data=csvread('Data.csv', 1, 1);
inputs=data(1:end-1,1:200)';
targets=data(1:end-1,end)';
targets(targets==0)=2;
targets=cevir1(targets,2)';

options = gaoptimset;
options = gaoptimset(options,'PopulationType', 'bitstring');
options = gaoptimset(options,'PopulationSize', 50);
options = gaoptimset(options,'EliteCount', 1);
options = gaoptimset(options,'CreationFcn', @gacreationuniform);
options = gaoptimset(options,'SelectionFcn',
@selectionstochunif);
options = gaoptimset(options,'CrossoverFcn',
@crossoversinglepoint);
options = gaoptimset(options,'MutationFcn', {  @mutationuniform
[] });

my_output= @(options,state,flag)
myoutput(options,state,flag,fid1);
myfit=@(x)myfuncnn(x,inputs,targets);
options = gaoptimset(options,'OutputFcns', my_output);
options = gaoptimset(options,'Display', 'iter');
options = gaoptimset(options,'PlotFcns', { @gaplotbestf });
[x,fval,exitflag,output,population,score] = ...
    ga(myfit,4,[],[],[],[],[],[],[],options);
s=int2str(x);
index=find(strncmp(s,dizi,length(s))==1);
score=cell2mat(scores(index))  ;
net=netler{index};net
mask=x;
masklog=logical(mask);
inputs=inputs(masklog,:);
outputs=net(inputs);
plotconfusion(targets,outputs);
```

```
%Yapay Sinir Ağları Fonksiyonu

function score=myfuncnn(x,inputs,targets)
    global dizi;
    global scores;
    global netler;


    s=int2str(x);

    if isempty(find(strncmp(s,dizi,length(s))==1, 1))

        if (all(x==0)==1)
            score=inf;
            dizi(length(dizi)+1)={s};
            scores(length(scores)+1)={score};
            hiddenLayerSize = 20;
            net = patternnet(hiddenLayerSize);
            netler(length(netler)+1)={net};
            return
        end

        mask=x;
        masklog=logical(mask);
        inputs=inputs(masklog,:);

        hiddenLayerSize = 20;
        net = patternnet(hiddenLayerSize);

        net.inputs{1}.processFcns =
{'removeconstantrows','mapminmax'};
        net.outputs{2}.processFcns =
{'removeconstantrows','mapminmax'};

        net.divideFcn = 'divideint';
        net.divideMode = 'sample';
        net.divideParam.trainRatio= 0.75;
        net.divideParam.valRatio =0.0;
        net.divideParam.testRatio = 0.25;

        net.trainFcn = 'trainlm';  % Levenberg-Marquardt

nnperformance
        net.performFcn = 'mse';

        net.plotFcns =
{'plotperform','plottrainstate','ploterrhist', ...
            'plotregression', 'plotfit'};

        net.trainParam.epochs=40;
        [net,tr] = train(net,inputs,targets);

        outputs = net(inputs);
        errors = gsubtract(targets,outputs);
        performance = perform(net,targets,outputs);
```

```
        trainTargets = targets .* tr.trainMask{1};
        valTargets = targets  .* tr.valMask{1};
        testTargets = targets  .* tr.testMask{1};
        trainPerformance = perform(net,trainTargets,outputs);
        valPerformance = perform(net,valTargets,outputs);
        testPerformance = perform(net,testTargets,outputs);
        score=performance;
        dizi(length(dizi)+1)={s};
        netler(length(netler)+1)={net};
        scores(length(scores)+1)={score};

    else
        index=find(strncmp(s,dizi,length(s))==1);
position index
        score=cell2mat(scores(index))  ;
    end
end
```

**Ek-I:**

```
% Genetik Algoritma Fonskiyonu
clear;
global dizi;
global scores;
global netler;

global divideParam;
global test;
global train;
global alldata;
global c1;
global goodtest goodtrain goodalldata;
global bestscore;
bestscore=1;

dizi={};
scores={};
netler={};

data=csvread('Data.csv', 1, 1);
inputs=data(1:end-1,1:200)';
targets=data(1:end-1,end)';
targets(targets==1)=2;
targets(targets==0)=1;

options = gaoptimset;
options = gaoptimset(options,'PopulationType', 'bitstring');
options = gaoptimset(options,'PopulationSize', 50);
options = gaoptimset(options,'EliteCount', 2);
options = gaoptimset(options,'CreationFcn', @gacreationuniform);
options = gaoptimset(options,'SelectionFcn',
@selectionstochunif);
options = gaoptimset(options,'CrossoverFcn',
@crossoversinglepoint);
```

```
options = gaoptimset(options,'MutationFcn', {  @mutationuniform
[] });

my_output= @(options,state,flag)
myoutput(options,state,flag,fid1);
myfit=@(x)myfuncCentroid(x,inputs,targets);
options = gaoptimset(options,'OutputFcns', my_output);
options = gaoptimset(options,'Display', 'iter');
options = gaoptimset(options,'PlotFcns', { @gaplotbestf });
[x,fval,exitflag,output,population,score] = ...
    ga(myfit,4,[],[],[],[],[],[],[],options);
s=int2str(x);
index=find(strncmp(s,dizi,length(s))==1);


train=goodtrain;
test=goodtest;
alldata=goodalldata;
bestscore
plotconfusion(train.targets2,train.outputs2,'train',test.targets2
,test.outputs2,'test',alldata.targets2,alldata.outputs2,'all');
metrics=metrik(alldata.targets',alldata.outputs')


%Centroid Tabanlı Algoritma Fonksiyonu

    global dizi;
    global scores;
    global netler;
    global divideParam;
    global test;
    global train;
    global alldata;
    global c1;
    global goodtest goodtrain goodalldata;
    global bestscore;


    s=int2str(x);

    if isempty(find(strncmp(s,dizi,length(s))==1, 1))

        if (all(x==0)==1)
            score=inf;
            dizi(length(dizi)+1)={s};
            scores(length(scores)+1)={score};


            return
        end

        mask=x;
        masklog=logical(mask);
        inputs=inputs(masklog,:);

divideParam.trainRatio= 0.75;
divideParam.valRatio =0.0;
```

```
divideParam.testRatio = 0.25;


nn = size(inputs,2);
ii = randperm(nn);
nt = round(nn*divideParam.trainRatio);
nv = round(nn*divideParam.valRatio);
nte = nn-nt-nv;
divideParam.trainInd = ii(1:nt);
divideParam.valInd = ii(nt+1:nt+nv);
divideParam.testInd = ii(nt+nv+1:nt+nv+nte);



train.inputs=inputs(:,divideParam.trainInd);
train.targets=targets(:,divideParam.trainInd);
test.inputs=inputs(:,divideParam.testInd);
test.targets=targets(:,divideParam.testInd);
c1.ind =(train.targets==1);
c2.ind=(train.targets==2);
c1.inputs=train.inputs(:,c1.ind);
c2.inputs=train.inputs(:,c2.ind);

c1.center=sum(c1.inputs,2).*(1/size(c1.inputs,2));
c2.center=sum(c2.inputs,2).*(1/size(c2.inputs,2));
c1.center=normc(c1.center); % normalize et
c1.center_test=repmat(c1.center,1,size(divideParam.testInd,2));
c2.center=normc(c2.center);
c2.center_test=repmat(c2.center,1,size(divideParam.testInd,2));


test.inputs_normalize=normc(test.inputs);
test.sonuc1=dot(c1.center_test,test.inputs_normalize,1);
test.sonuc2=dot(c2.center_test,test.inputs_normalize,1);
test.outputs=test.sonuc1>test.sonuc2;
test.outputs=double(test.outputs);
test.outputs(test.outputs==0)=2;
test.outputs2=cevir1(test.outputs,2)';
test.targets2=cevir1(test.targets,2)';

c1.center_train=repmat(c1.center,1,size(divideParam.trainInd,2));
c2.center_train=repmat(c2.center,1,size(divideParam.trainInd,2));

train.inputs_normalize=normc(train.inputs);
train.sonuc1=dot(c1.center_train,train.inputs_normalize,1);
train.sonuc2=dot(c2.center_train,train.inputs_normalize,1);
train.outputs=train.sonuc1>train.sonuc2;
train.outputs=double(train.outputs);
train.outputs(train.outputs==0)=2;
train.outputs2=cevir1(train.outputs,2)';
train.targets2=cevir1(train.targets,2)';

c1.center_all=repmat(c1.center,1,size([divideParam.trainInd
divideParam.testInd],2));
c2.center_all=repmat(c2.center,1,size([divideParam.trainInd
divideParam.testInd],2));
```

```
alldata.inputs_normalize=normc(inputs);
alldata.sonuc1=dot(c1.center_all,alldata.inputs_normalize,1);
alldata.sonuc2=dot(c2.center_all,alldata.inputs_normalize,1);
alldata.outputs=alldata.sonuc1>alldata.sonuc2;
alldata.outputs=double(alldata.outputs);
alldata.outputs(alldata.outputs==0)=2;
alldata.outputs2=cevir1(alldata.outputs,2)';
alldata.targets=targets;
alldata.targets2=cevir1(targets,2)';

         score=1-metrics(1)      1-metrics(1) is misfit

        dizi(length(dizi)+1)={s};
        scores(length(scores)+1)={score};

        if(score<bestscore)
         bestscore=score;
        goodtrain=train;
        goodtest=test;
        goodalldata=alldata;
        end
    else
        index=find(strncmp(s,dizi,length(s))==1);
position index
        score=cell2mat(scores(index))  ;
      end

end
```

## ÖZGEÇMİŞ

**Ad-Soyad** **:** İlkay YELMEN
**Doğum Tarihi ve Yeri** **:** 15.11.1989, Çankırı
**E-posta** **:** ilkayyelmen@hotmail.com

## ÖĞRENİM DURUMU:

- **Lisans : 2013**, İstanbul Aydın Üniversitesi, Mühendislik Mimarlık Fakültesi, Yazılım Mühendisliği Bölümü
- **Yüksek Lisans : 2015-Devam Ediyor**, İstanbul Aydın Üniversitesi, Bilgisayar Mühendisliği A.B.D., Bilgisayar Mühendisliği Programı

## TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- Yelmen İ. Zontul M. (2016) "Sentiment Analysis Tool for Daily Speech Turkish Texts", *International Journal of Research in Engineering and Technology (IJRET)* Vol. 4, No. 1, 2016

## DİĞER YAYINLAR, SUNUMLAR VE PATENTLER:

- Yelmen İ. Tosyalıoğlu N. (2013) "Evaluation of the University Students' Opinions on Environmental Awareness Using Data Mining Association Rule" *International Journal of Electronics, Mechanical and Mechatronics Engineering,* Vol.2, Num.4, pp.384-390.

- Yelmen İ. Zontul M. (2016) "Determining The Core Part of Software Development Curriculum Applying Association Rule Mining on Software Job Ads In Turkey" *Fourth International Conference on Data Mining & Knowledge Management Process (DKMP 2016)*