



**ETKİN SINIFLANDIRMA İÇİN GENETİK ALGORİTMA TABANLI
ÖZNİTELİK ALT KÜME SEÇİMİ**

Shima AFZALI VAHED MOGHADDAM

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2014

Shima AFZALİ VAHED MOGHADDAM tarafından hazırlanan “ETKİN SINIFLANDIRMA İÇİN GENETİK ALGORİTMA TABANLI ÖZNİTELİK ALT KÜME SEÇİMİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi BİLGİSAYAR MÜHENDİSLİĞİ Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Oktay YILDIZ

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

.....

Başkan : Doç. Dr. Diyar AKAY

Endüstri Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

.....

Üye : Yrd. Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum

.....

Tez Savunma Tarihi: 26/12/2014

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Şeref SAĞIROĞLU

Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Shima AFZALI VAHED MOGHADDAM

31/12/2014

ETKİN SINIFLANDIRMA İÇİN GENETİK ALGORİTMA TABANLI ÖZNİTELİK
ALT KÜME SEÇİMİ
(Yüksek Lisans Tezi)

Shima AFZALI VAHED MOGHADDAM

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Aralık 2014

ÖZET

Sınıflandırma, Makine Öğrenmesi ve Veri Madenciliği alanında sıklıkla kullanılan önemli yöntemlerden biridir. Öznitelik seçme teknikleri ise sınıflandırma başarısını ve performansını arttırmak için son zamanlarda sıklıkla çalışılan bir alandır. Öznitelik seçimi, metin madenciliği, biyoinformatik ve görüntü analizi gibi pek çok alanda yaygın bir şekilde uygulanmaktadır. Öznitelik seçme teknikleri veri kümesi içinden, sınıflandırma başarısını etkileyen ilgisiz niteliklerin atılması ya da önemli niteliklerin seçilmesi şeklinde olabilir. Böylece sınıflandırıcıların başarısı artırılabilir. Bu çalışmada, genetik algoritma tabanlı hibrit etkin bir öznitelik seçme metodu önerilmektedir. Bu metot, literatürde sınıflandırma amaçlı sıklıkla kullanılan dört veri kümesi üzerinde denemiştir. Önerilen metot iki aşamadan oluşmaktadır. İlk aşamada, öznitelik havuzu oluşturmak için öznitelik sıralama metotları kullanılmıştır. İkinci aşamada, yüksek sınıflandırma performansı ile en uygun öznitelik alt kümesini seçmek için genetik algoritma kullanılmıştır. Bu çalışmada genetik algoritma ve dört ayrı sınıflandırma algoritması kullanılmıştır. Önerilen metot UCI'dan elde edilen Meme kanseri wisconsin teşhis (WDBC), Tek proton emisyon bilgisayarlı tomografi (SPECT) Heart, Statlog heart ve Meme kanseri wisconsin prognostik (WPBC) veri kümelerinde denenmiş, sırayla 100%, 91,25%, 96,29%, ve 94,8276% sınıflandırma başarısı ile etkin bir öznitelik seçme gerçekleştirilebilmiştir.

Bilim Kodu : 902.1.104

Anahtar Kelimeler : Sınıflandırma, öznitelik seçimi, genetik algoritma, makine öğrenmesi, veri madenciliği

Sayfa Adedi : 111

Danışman : Öğr. Gör. Dr. Oktay Yıldız

FEATURE SUBSET SELECTION METHOD FOR AN EFFECTIVE CLASSIFICATION
BASED ON GENETIC ALGORITHM

(M. Sc. Thesis)

Shima AFZALI VAHED MOGHADDAM

GAZI UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

December 2014

ABSTRACT

Classification is one of the important methods commonly used in the Machine Learning and Data Mining. Feature selection technique has frequently used to improve the classification performance in the last decade. Feature selection has applied in wide variety of real world applications, such as text mining, bioinformatics and image analysis. Feature selection techniques within the data set can be classified as disposal of irrelevant attributes that adversely affect performance or the selection of important attributes. Thus, the performance of classifiers can be increased. In this study, a hybrid effective feature selection method is proposed based on genetic algorithm. This method was applied on 4 data sets that have commonly used in the literature. The proposed method consists of two steps. In the first stage, feature ranking methods were used to create feature pool. In the second stage, a genetic algorithm was used to select the proposed optimal subset of features with high classification performance. In this study, the genetic algorithm and four other different classification algorithms were used. The proposed method was applied on wisconsin diagnostic breast cancer (WDBC), Single proton emission computed tomography (SPECT) heart, Statlog heart and Wisconsin prognostic breast cancer (WPBC) data sets taken from the UCI and achieved effective feature selection with 100%, 91,25%, 96,29%, and 94,8276% classification accuracy, respectively.

Science Code : 902.1.104

Key Words : Classification, feature selection, genetic algorithm, machine learning, data mining

Page Number : 111

Supervisor : Lect.Dr. Oktay YILDIZ

TEŞEKKÜR

Çalışmalarım boyunca değerli yardımlarıyla ve katkılarıyla beni yönlendiren Sayın hocam Dr. Oktay YILDIZ'a; manevi katkı ve desteğinden dolayı aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	xii
SİMGELER VE KISALTMALAR.....	xiv
1. GİRİŞ.....	1
2. LİTERATÜR.....	5
3. BOYUT İNDERGEME	7
3.1. Öznitelik Seçimi.....	8
3.1.1. Öznitelik sıralama	9
3.1.2. Öznitelik alt küme seçimi.....	10
4. ÖZNİTELİK ALT KÜME SEÇİMİ	13
4.1. Filtreleme Yöntemi	13
4.1.1. Tek değişkenli	14
4.2. Sarmalama Yöntemi	15
4.2.1. Deterministik.....	16
4.2.2. Rastasal	17
4.3. Gömülü Yöntemi.....	22
5. SINIFLANDIRMA	25
5.1. Naive Bayes.....	25

	Sayfa
5.2. Destek Vektör Makinesi.....	26
5.2.1. Doğrusal destek vektör makineleri.....	27
5.2.2. Doğrusal olmayan destek vektör makineleri.....	30
5.3. Karar Ağacı	32
5.4. K En Yakın Komşu (KEYK)	34
5.5. Doğrusal Ayırıcı Analizi	35
5.6. Sınıflandırıcı Performansı	36
6. GERÇEKLEŞTİRİLEN ÇALIŞMA.....	39
6.1. Kullanılan Veri Kümeler	39
6.1.1. Meme kanseri veri kümesi (WDBC).....	39
6.1.2. SPECTF kalp veri kümesi	40
6.1.3. Meme kanseri wisconsin (Prognostic) veri kümesi (WPBC)	40
6.1.4. Statlog (kalp) veri kümesi	41
6.2. Deneysel Çalışma.....	42
6.2.1. Birinci aşama: Öznitelik havuzu oluşturulması	44
6.2.2. İkinci aşama: Genetik algoritma ile özniteliklerin belirlenmesi	57
6.3. Sınıflandırma.....	60
6.3.1. WDBC veri kümesi	60
6.3.2. SPECTF veri kümesi.....	68
6.3.3. Statlog kalp veri kümesi.....	77
6.3.4. WPBC veri kümesi.....	86
7. SONUÇ VE ÖNERİLER.....	97
KAYNAKLAR	103
ÖZGEÇMİŞ	111

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Arama modellerin avantajları, dezavantajları ve örnekler [16]	22
Çizelge 6.1. UCI veri tabanından alınan veri kümeleri	39
Çizelge 6.2. WDBC veri kümesi nitelikleri için FDR değeri	44
Çizelge 6.3. WDBC veri kümesinin niteliklerinin sıralanmış FDR değerleri	45
Çizelge 6.4. WDBC veri kümesi nitelikleri için Relief değerleri	45
Çizelge 6.5. WDBC veri kümesinin niteliklerinin sıralanmış Relief değerleri	46
Çizelge 6.6. WDBC veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri	46
Çizelge 6.7. SPEFCT veri kümesi nitelikleri için FDR değeri	48
Çizelge 6.8. SPEFCT veri kümesinin niteliklerinin sıralanmış FDR değerleri	48
Çizelge 6.9. SPEFCT veri kümesi nitelikleri için Relief değerleri.....	49
Çizelge 6.10. SPEFCT veri kümesinin niteliklerinin sıralanmış Relief değerleri	50
Çizelge 6.11. SPEFCT veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri	50
Çizelge 6.12. Statlog veri kümesi nitelikleri için FDR değeri	52
Çizelge 6.13. Statlog veri kümesinin niteliklerinin sıralanmış FDR değerleri	52
Çizelge 6.14. Statlog veri kümesi nitelikleri için Relief değerleri.....	52
Çizelge 6.15. Statlog veri kümesinin niteliklerinin sıralanmış Relief değerleri	52
Çizelge 6.16. Statlog veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri	53
Çizelge 6.17. WPBC veri kümesi nitelikleri için FDR değeri	54
Çizelge 6.18. WPBC veri kümesinin niteliklerinin sıralanmış FDR değerleri	54
Çizelge 6.19. WPBC veri kümesi nitelikleri için Relief değerleri.....	55
Çizelge 6.20. WPBC veri kümesinin niteliklerinin sıralanmış Relief değerleri	55

Çizelge	Sayfa
Çizelge 6.21. WPBC veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri	56
Çizelge 6.22. WDBC veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu	60
Çizelge 6.23. WDBC veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu	62
Çizelge 6.24. WDBC veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu	64
Çizelge 6.25. WDBC veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu	66
Çizelge 6.26. WDBC veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi.....	67
Çizelge 6.27. SPECTF veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu	69
Çizelge 6.28. SPECTF veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu	71
Çizelge 6.29. SPECTF veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu	73
Çizelge 6.30. SPECTF veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu	74
Çizelge 6.31. SPECTF veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi.....	76
Çizelge 6.32. Statlog veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu	78
Çizelge 6.33. Statlog veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu	80
Çizelge 6.34. Statlog veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu	82
Çizelge 6.35. Statlog veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu	83
Çizelge 6.36. Statlog veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi.....	85

Çizelge	Sayfa
Çizelge 6.37. WPBC veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu	87
Çizelge 6.38. WPBC veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu	89
Çizelge 6.39. WPBC veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu	91
Çizelge 6.40. WPBC veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu	93
Çizelge 6.41. WPBC veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi.....	94
Çizelge 7.1. WDBC veri kümesinin Orijinal 30 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması.....	98
Çizelge 7.2. SPECTF veri kümesinin orijinal 44 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması.....	98
Çizelge 7.3. Statlog veri kümesinin orijinal 13 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması	99
Çizelge 7.4. WPBC veri kümesinin orijinal 32 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması	99
Çizelge 7.5. WDBC veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması	100
Çizelge 7.6. WPBC veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması	100
Çizelge 7.7. Spectf veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması	101
Çizelge 7.8. Statlog veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması	101

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Bir öznitelik seçimi sürecinin bir birleşik görünümü.	9
Şekil 3.2. Doğrulama ile öznitelik seçim süreci.	12
Şekil 4.1. Relief için sözde kod [66].....	15
Şekil 4.2. GA için sözde kod [72].....	18
Şekil 4.3. Rulet tekerliği seçimi için sözde kod.....	19
Şekil 5.1. Marjini maksimize eden ayırma düzlemi [75].....	27
Şekil 5.2. Doğrusal ayrılabilir durum için alt düzlem ayırımı [75].....	28
Şekil 5.3. Hata matrisi.....	37
Şekil 6.1. WDBC veri kümesi nitelik ve açıklamaları.....	40
Şekil 6.2. WPBC veri kümesi nitelik ve açıklamaları.	41
Şekil 6.3. Statlog veri kümesi nitelik ve açıklamaları.	42
Şekil 6.4. Genetik algoritma tabanlı etkin öznitelik seçimi.	43
Şekil 6.5. WDBC veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması.....	47
Şekil 6.6. SPECTF veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması.....	51
Şekil 6.7. Statlog veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması.....	53
Şekil 6.8. WPBC veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması.....	56
Şekil 6.9. GA modelinin blok diyagramı	57
Şekil 6.10. Çaprazlama işlevi.....	59
Şekil 6.11. Mutasyon işlemi	59
Şekil 6.12. WDBC veri kümesinde DVM sınıflandırıcının hata matrisi	61
Şekil 6.13. WDBC veri kümesinde DAA sınıflandırıcının hata matrisi.....	62

Şekil	Sayfa
Şekil 6.14. WDBC veri kümesinde NB sınıflandırıcının hata matrisi	64
Şekil 6.15. WDBC veri kümesinde KA sınıflandırıcının hata matrisi.....	66
Şekil 6.16. WDBC veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları	68
Şekil 6.17. SPCTF veri kümesinde DVM sınıflandırıcının hata matrisi	69
Şekil 6.18. SPCTF veri kümesinde DAA sınıflandırıcının hata matrisi	71
Şekil 6.19. SPCTF veri kümesinde NB sınıflandırıcının hata matrisi	73
Şekil 6.20. SPCTF veri kümesinde KA sınıflandırıcının hata matrisi	75
Şekil 6.21. SPECTF veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları	77
Şekil 6.22. Statlog veri kümesinde DVM sınıflandırıcının hata matrisi.....	78
Şekil 6.23. Statlog veri kümesinde DAA sınıflandırıcının hata matrisi	80
Şekil 6.24. Statlog veri kümesinde NB sınıflandırıcının hata matrisi.....	82
Şekil 6.25. Statlog veri kümesinde KA sınıflandırıcının hata matrisi	84
Şekil 6.26. Statlog veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları	86
Şekil 6.27. WPBC veri kümesinde DVM sınıflandırıcının hata matrisi.....	87
Şekil 6.28. WPBC veri kümesinde DAA sınıflandırıcının hata matrisi	89
Şekil 6.29. WPBC veri kümesinde NB sınıflandırıcının hata matrisi.....	91
Şekil 6.30. WPBC veri kümesinde KA sınıflandırıcının hata matrisi	93
Şekil 6.31. WPBC veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları	95

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
CFS	Correlation based feature selection
CART	Classification And Regression Trees
DAA	Doğrusal ayırıcı analizi
DVM	Destek vektör makineleri
EDA	Estimation of distribution algorithms
FCBF	Fast Correlation Based Filter
FDR	Fisher discriminant ratio
FS	Feature selection
GA	Genetik algoritma
GÖS	Geriye öznitelik seçimi
İÖS	İleri öznitelik seçimi
IBL	Instance based learning
ID	İterative Dichotomiser
KEYK	K-En Yakın Komşu
KAÖPİ	Karar Ağaçları Ölçeklenebilir Paralelleştirilebilir İndüksiyon
KA	Karar Ağacı
MBF	Markov blanket filtre
NB	Naive Bayes
RTF	Radyal Taban Fonksiyonu
SMO	Sıralı Minimal Optimizasyonu
SPECT	Single Proton Emission Computed Tomography
TB	Tavlama benzetimi
TBA	Temel bileşen analizi
UCI	University of California, School of Information and Computer Science

Kısaltmalar**Açıklamalar****WDBC**

Wisconsin diagnostic breast cancer

WPBC

Wisconsin pronostic breast cancer

YSA

Yapay sinir ağı

1. GİRİŞ

Sınıflandırma, Makine Öğrenmesi ve Veri Madenciliği alanında sıklıkla kullanılan önemli yöntemlerden biridir. Sınıflandırma algoritmaları, önceden bilinen, etiketli verileri eğitim verisi olarak kullanır. Literatürde pek çok sınıflandırma algoritması önerilmiştir. Destek Vektör Makinesi, Naive Bayes, Karar Ağaçları, Yapay Sinir Ağları bunlar içinde öne çıkan bazı sınıflandırma algoritmalarıdır.

Sınıflandırma algoritmalarının kullandıkları eğitim verileri, örneklerden oluşur. Her örneğin sahip olduğu çeşitli özellikler (nitelikler) vardır. Her bir nitelik, sınıflandırma algoritmasının başarısında aynı etkiye sahip olmayabilir. Hatta bazı nitelikler ilgisiz olarak adlandırılan ve sınıflandırma başarısını veya performansını olumsuz etkileyen nitelikler olabilir [1]. Literatürde öz nitelik seçme olarak adlandırılan çalışmalar yaygın bir şekilde yer almaktadır.

Sınıflandırma, istatistik, makine öğrenmesi, veri madenciliği gibi alanlarda çeşitli sınıflandırma yöntem ve teknikleri mevcuttur. Aynı tür problemlerde farklı sınıflandırma algoritmaları farklı sonuçlar verebilmektedir. Elde edilen öz nitelik vektörlerine en uygun sınıflandırıcıyı seçebilmek için sınıflandırma yöntemlerinin özelliklerinin iyi bilinmesi gerekmektedir.

Öz nitelik seçme teknikleri ise sınıflandırma başarısını ve performansını arttırmak için son zamanlarda sıklıkla çalışılan bir alandır. Öz nitelik seçimi, metin madenciliği, biyoinformatik ve görüntü analizi gibi pek çok alanda yaygın bir şekilde uygulanmaktadır. Öz nitelik seçme teknikleri veri kümesi içinden, sınıflandırma başarısını etkileyen ilgisiz niteliklerin atılması ya da önemli niteliklerin seçilmesi şeklinde olabilir. Böylece sınıflandırıcıların başarısı artırılabilir.

Veri kümesindeki ilgisiz ve fazlalık olan nitelikler bir sistemin performansını olumsuz şekilde etkileyebilir. Aynı zamanda, bazı durumlarda, daha az öz niteliklere sahip daha etkili sistemler olması arzu edilmektedir. Bu yüzden öz nitelik seçimi algoritmaları genellikle bu hedef için kullanılan yöntemlerdir [2].

Öznitelik seçimi, önemli bir makine öğrenme yöntemidir. Verimli ve yaygın bir şekilde sınıflandırma sistemlerinde kullanılmaktadır. Ayırt edici özniteliklerin seçimi, tanıma başarısını arttırabilmektedir. Bunun yanında öznitelik seçiminin birçok avantajı bulunmaktadır. Seçilen özniteliklerle yapılan sınıflandırmada işlem sayısı daha azdır, gürültülü ve ilgisiz öznitelikler özgün veriden çıkarılarak sınıflama başarısı arttırılır, öznitelikler üzerinden yapılabilen sınıflama yorumları artar veya kolaylaşır. Eğitim zamanı kısılır, daha az ölçüm yapılır ve daha az bellek kullanılır. Bunlar, anlamlı ve daha kolay sınıflandırma sağlar [3-4].

Öznitelik seçimi ve boyut indirgeme için temel bileşen analizi, faktör analizi gibi birçok yöntem bulunmaktadır. Boyut indirgeme için kullanılan yöntemlerden bazıları özgün öznitelikler üzerinden yapılan dönüşümlerle yeni özniteliklerin bulunmasıdır ki, bu durum özgün verinin kaybolmasına neden olur. Örneğin temel bileşen analizi, tekil değer ayrıştırması ve faktör analizleri özgün veriyi yeni bir veri uzayına taşırlar [5]. Bazı durumlarda özgün verinin olduğu gibi korunması, çıkan sınıflama sonuçlarının yorumlanabilmesi için önemlidir. Bu nedenle verinin korunarak içlerinden en iyi özniteliklerin seçilmesi üzerine çalışmalar da yapılmaktadır [3-4].

Öznitelik seçimi algoritmaları, öğrenme algoritmasına bağımlılığına dayalı, üç ana gruba ayrılır: Filtreleme, sarmalama ve gömülü yöntemler. Birçok örüntü tanıma tekniği, alakasız veya fazlalıklara neden olan özniteliklerin büyük miktarı ile başa çıkmak için tasarlanmamışlardır. Öznitelik seçimi teknikleri ile birleştirilerek pek çok uygulamada bir gereklilik haline gelmiştir.

Öznitelik seçimi pek çok uygulama alanlarında gereklidir. Örneğin, yazılım ölçümleri analizi [6], metin madenciliği [7], gen ifade ve mikrodizi analizi [8], görüntü analizi [9], web madenciliği [10] ve saldırı tespit sistemleri [11].

Bu çalışmada filtreleme ve sarmalama yöntemleri birlikte kullanarak hibrit bir öznitelik seçme modeli sunulmuştur. Bu amaçla, her iki yöntem de denenerek, en güvenli ve daha etkin bir öznitelik alt kümesi çıkarmak için yeni bir yöntem önerilmiştir.

Burada, öznitelik seçimi için Fisher Ayırım Oranı (FDR) ve Relief Algoritması, Genetik Algoritma ile birlikte kullanılmıştır. Sınıflandırma doğruluğunu karşılaştırmak için dört

farklı sınıflandırma yöntemi (Destek Vektör Makineleri (DVM), Naive Bayes (NB), Doğrusal Ayırma Analizi (DAA) ve Karar Ağacı (KA)) kullanılmıştır.

Gerçekleştirilen çalışma iki aşamadan oluşmaktadır. İlk aşamada, FDR ve Relief öznitelik sıralama yöntemleri uygulanmış, ikinci aşamada GA ile etkin nitelikler belirlenmiştir. Böylece orijinal öznitelik boyutu indirilerek GA için arama uzayı daraltılmıştır. Bu nedenle, GA, kısa sürede yüksek doğruluk ile etkili ve güçlü bir öznitelik alt kümesi oluşturabilir. İkinci aşamada, GA yöntemi dört farklı sınıflandırma yöntemi için (DVM, NB, DAA ve KA) denenmiştir. Önerilen hibrit yapının amacı, sınıflandırma performansını maksimize edip, aynı zamanda öznitelik alt küme boyutunu en aza indirmektir. Aslında, bu yaklaşım yüksek boyuta sahip olan tüm veri setleri için de geçerlidir.

Bu çalışmada, eğitim ve test deneyleri için Irvine California Üniversitesi (UCI) makine öğrenme deposundan alınan Wisconsin meme kanseri (WDBC), kalp tek proton emisyon hesaplanan tomografi (SPECTF), Statlog ve Wisconsin prognostik meme kanseri tanısı (WPBC) veri kümeleri kullanılmıştır [12]. Sınıflandırma doğruluğu, öznitelik küme boyutu, eğitim ve test küme oranı değerlendirildiğinde deneysel sonuçlar bu çalışmanın önceki çalışmalardan daha iyi çalıştığını göstermektedir.

2. LİTERATÜR

Öznitelik seçimi ile ilgili literatürde birçok araştırma olmakla birlikte [13-15], en iyi sınıflandırma başarısı için öznitelik seçme algoritmalarını karşılaştıran pek çok çalışma da yapılmaktadır [16-19].

Son zamanlarda, öznitelik seçimi araştırmaları çeşitli nedenlerle artış göstermiştir. Bunun nedeni, veri madenciliği [20-21], tıbbi veri işleme [22] ve multimedya bilgi alma [23-25] gibi büyük miktarda veri ile ilgilenen yeni uygulamalar geliştirilmiş olmasıdır.

Öznitelik seçme yöntemlerinde, genetik algoritma (GA) oldukça önemli bir yer tutar. GA doğal evrim sürecinden esinlenmiş bir yöntemdir. Doğal gelişimi taklit eden birçok mekanizmaya sahiptir [26]. Bilimsel, mühendislik optimizasyonu veya arama problemlerindeki potansiyeli, büyüktür. Ayrıca, GA öznitelik seçimi için doğal olarak uygulanabilir. Siedlecki ve Sklansky çalışması, klasik algoritmaların temsilcisine göre GA'nın üstünlüğünü gösteren ilk çalışmaların örneğidir [27]. Daha sonra, öznitelik seçimi için GA'nın avantajlarını gösteren birçok çalışma yayınlanmıştır [28-31].

Basit veya tek bir GA ile ilgili sınırlamalar birçok uygulamada ele alınmıştır. Normal şartlar altında, basit bir GA tarafından sağlanan çözümler, klasik sezgisel algoritmalara göre daha iyi sonuçlar elde etmeyebilir. Pratik ve etkin bir şekilde bu sınırlamayı aşmak için GA hibritize edilmektedir. Hibritize etmenin üç yolu [32]'de önerilmiştir. Bunlar, problem-özel kodlamada, özel genetik operatörlerin kullanımı ve klasik algoritmaların iyi özelliklerini dâhil etmektir. Bu ilkeleri takip eden hibrit GA çeşitli uygulama alanlarında geliştirilmiş ve başarılı performans elde edilmiştir [33-35].

Özellikle eksik veya hatalı veri içeren veri kümelerinde zorluklarla karşılaşmaktadır. Bu veriler eksik (eksik parametre değerleri), yanlış (verilerde sistematik veya rasgele gürültü), seyrek (az ve/veya gösterilmeyen kayıtların olması) ve hatalı (verilen bir görev için parametrelerin uygunsuz seçimi) bilgi içerebilir. Makine öğrenmesi, bu amaçla pek çok araç sağlar.

Saeys ve ark. [36], öznitelik seçiminin sınıflandırma için gerekliliğini bahsetmiştir. Biyoinformatik alanında sınıflandırma için farklı öznitelik seçme tekniklerinden bahsetmiştir. Yüksek boyutlu veri, birçok sınıflandırma algoritması için ciddi bir sorundur ve ayrıca hesaplama maliyetinin artmasına ve bellek kullanımının yükselmesine sebep olur. Tan ve ark. [37], boyut indirgeme ile daha anlaşılabilir modeller elde etmek, farklı görüntüleme tekniklerinin kullanımını basitleştirmesi üzerine çalışmalar yapmışlardır. Molina ve ark. [38], Guyon ve Elisseeff [39, 40] çalışmalarında, ilgili ve ilgisiz niteliklerin, sentetik veri kümelerinde etkilerini göstermişlerdir. Bu hedefe ulaşmak için, üretilen veri setleri kullanılarak yapay yollarla kontrollü deney seti tasarlanmıştır.

Dash ve Liu [17] tipik bir öznitelik seçimi sürecinin dört adımını anlatmışlar. Bunlar: nesil (üretim) prosedürü, değerlendirme fonksiyonu, kriter durdurma ve doğrulama prosedürü olarak adlandırılmaktadır. Bu çalışmada, nesil prosedürü üç kategoriye ayrılmıştır: komple, sezgisel ve rastgele; değerlendirme fonksiyonları beş kategoride yer almıştır: mesafe, bilgi, bağımlılık, tutarlılık ve sınıflandırıcı hata oranı ölçüleri. Genetik algoritmalar aynı zamanda yaygın bir şekilde görüntü işleme ve örüntü tanıma alanlarında kullanılmaktadır [41-43]. Genetik algoritmalar otomatik olarak birçok farklı özniteliklerin alaka (ilgili/relative) önemini belirlemek ve sistem için özniteliklerin bir iyi alt kümesini seçmek için kullanılır [44]. Matsui ve ark. [45] beyin gri / beyaz madde bölgelerini sınıflandırmak için yapay sinir ağlarını (YSA) kullanmışlardır. YSA'nın sınıflandırma performansını artırmak amacıyla öznitelikleri belirlemişler, bunun için GA kullanmışlardır. Emmanouilidis ve ark. [46] öznitelik seçimi için çok-kriterli GA kullanımını tartışmaktadırlar. GA, çok-kriterli uygunluk fonksiyonlar ile yüksek sınıflandırma doğruluğunu korurken, seçilen özniteliklerin sayısını en aza indirmeye çalışır. Algoritma çeşitli doğruluk ve trade-off karmaşıklığı ile alternatif öznitelik alt kümelerinin çeşitli bir popülasyonunu elde etmek için gösterilmiştir. Bu algoritma, bulanık modeller ile sınıflandırmayı gerçekleştirmek amacıyla öznitelikleri seçmek için uygulanır. Her veri noktasında dokuz giriş niteliği ve bir sonuç etiketine sahip (kötü huylu veya iyi huylu) kanser veri üzerinde denenmiştir.

3. BOYUT İNDİRGE ME

Bir bilgisayar de var olan her veri, niteliklerin bir dizisi olarak kayıt edilir ve gösterilir. Örneğin, yaş, boy, kilo vb. nitelikler değiştirilebilir şekilde, boyut olarak adlandırılabilir. Böylece n niteliğe sahip olan bir varlık, bir n -boyutlu uzayda çok boyutlu bir nokta olarak gösterilebilir. Bir nitelik kümesinin indirgeme sürecinde ilk n nitelikten oluşan nitelik kümesi bir k nitelikten oluşan bir öznitelik kümesine indirgenir ($k < n$) ki buna boyut indirgeme denilir. İdeal olarak, k azalmış öznitelik, orijinal n niteliğin önemli özelliklerini taşır.

Veri tabanı ve makine öğrenme sistemleri gibi birçok alanda boyut indirgeme önemli bir rol oynar ve dolayısıyla bir veri indirgeme, daha iyi veri görselleştirme, gelişmiş sınıflandırma doğruluğu, hızlı ve verimli veri alma, indeks performansı artırma [47-48] gibi çok değerli sonuçlar sunar. Boyut indirgeme teknikniği iki önemli kategoriye ayrılır: öznitelik çıkarma ve öznitelik seçimi.

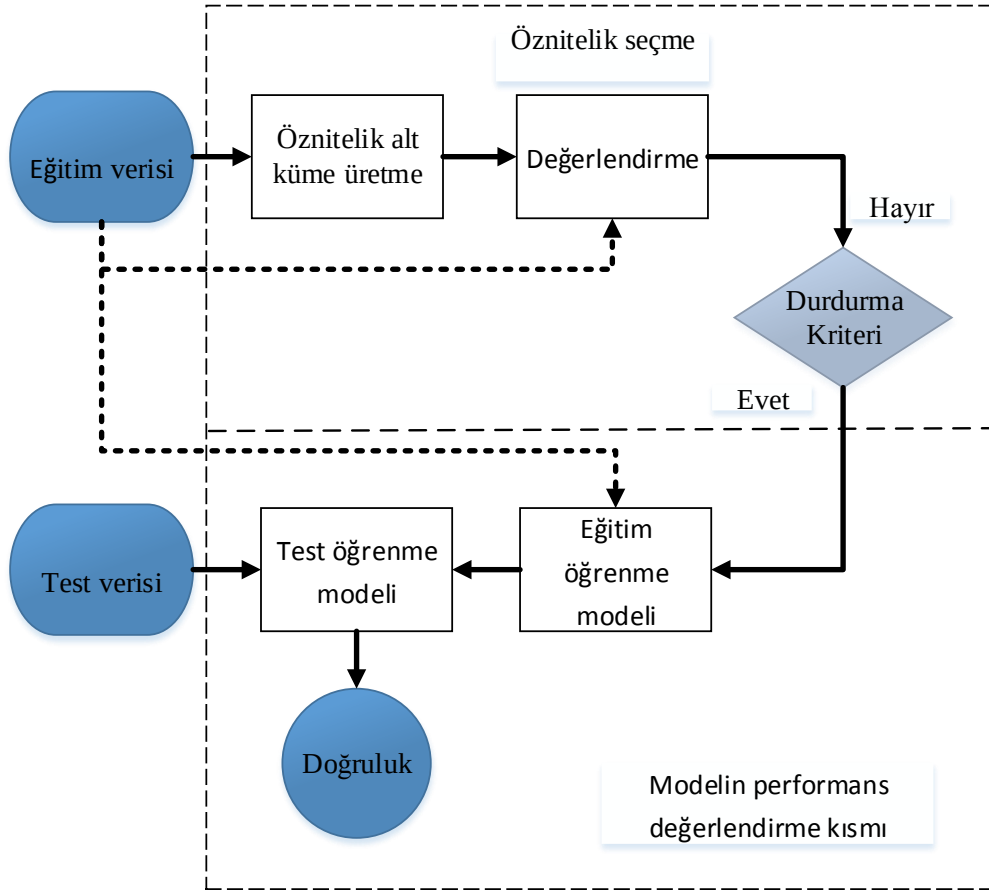
Öznitelik çıkarma, öznitelik dönüşümü olarak tanımlanır. n orijinal boyutlu bir kombinasyonda bir yeni k boyutlu nitelik bulma işlemidir. En iyi bilinen öznitelik çıkarma teknikleri, projeksiyon ve sıkıştırma yöntemlerine dayanmaktadır. Temel bileşenler analizi (TBA) ve doğrusal ayırıcı analizi (DAA) sırasıyla denetimsiz ve denetimli öğrenme için projeksiyon yöntemleri örnekleridir. Karşılıklı bilgi ve bilgi teorisi, sıkıştırma yönteminde kullanılmaktadır [49-51].

Öznitelik çıkarma yöntemin aksine, öznitelik seçimi n niteliğin orijinal setinden k tane en iyi öznitelikten oluşan bir alt küme elde etmeyi amaçlar ve geri kalan nitelikleri göz ardı eder. Öznitelik seçme tekniği, niteliklerin orijinal temsilini değiştirmez [52].

3.1. Öznitelik Seçimi

Öznitelik seçimi (aynı zamanda nitelik seçimi veya değişken seçimi olarak belirlenen) bazı kriterlere göre orijinal giriş niteliklerden bir uygun (optimum) niteliklerin alt kümesini seçmek için geliştirilmiş bir tekniktir. Niteliklerin bir alt kümesinin bulma nedeni; daha düşük boyutta bir sorunun çözümünün her zaman daha kolay olmasıdır. Giriş ve çıkış değişkenleri arasında doğrusal olmayan eşleştirmenin anlaşılmasında bize yardımcı olur [53]. Öznitelik seçimi, belli bir büyüklükteki niteliklerin en uygun alt kümesini bulma sürecidir, bu durum mümkün olan en büyük genellemeyi sağlar [54].

Şekil 3.1, öznitelik seçim sürecini açıklamaktadır. Öznitelik seçme süreci üç yönlüdür. İlk olarak, boyut indirgeme işlevi ile sınıflandırıcının tahmin doğruluğunu artırmak. Bu adım dikkat çekici bir şekilde bir sınıflandırıcının öngörü doğruluğunu artırabilir. İkinci olarak, hesaplama maliyetini düşürür. Öğrenme algoritmaların birçoğu eğitim ve tahmin adımlarında özniteliklerin sayısı büyük olduğunda hesaplama açısından zorluk yaşar. Eğitim algoritmasında, önce öznitelik seçimi adımı hesaplama yükünü azaltabilir. Son olarak, boyut indirgeme, veri üretme sürecine daha iyi bakış açısı sağlar. Bu önemlidir, çünkü birçok durumda bilgilendirici öznitelikleri belirtme yeteneği önemlidir [39]. Özniteliklerin seçimi iki şekilde elde edilebilir: öznitelik sıralama ve öznitelik alt küme seçimi [39, 55].



Şekil 3.1. Bir öznitelik seçimi sürecinin birleşik görünümü

3.1.1. Öznitelik sıralama

Öznitelik sıralama yaklaşımı, niteliklerin bireysel ilgisine göre gerçekleştirilir. Bu teknikte, bazı kriterlere göre üst n sıradaki nitelikler seçilmek için sıralanır ve bu n sayısı kullanıcı [56] tarafından veya otomatik olarak belirlenir [57]. Bu yöntemin dezavantajı, öznitelikleri birbirinden bağımsız olarak varsaymasıdır. Bu iki soruna neden olabilir [58]: Bireysel olarak ilgili olmamak nedeniyle atılan nitelikler, bazı diğer nitelikler ile değerlendirildiğinde, ilgi katsayısı artabilir. Bireysel olarak ilgili olarak kabul edilen nitelikler, gereksiz veya fazlalığa neden olabilirler.

Öznitelik sıralamada, puanlama fonksiyonları (Öklid uzaklık), korelasyon (Pearson korelasyon katsayısı) veya bilgi tabanlı ölçütleri değerlendirme kriteri olarak kullanılabilir. Genellikle bunlar ön işleme adımları olarak kullanılır çünkü hesaplama bakış açısından çok verimlidir [59]. Bu yöntemde, beklenen öznitelikler, sadece öznitelik kombinasyonun hedef fonksiyona uymaması durumunda başarısız olur.

3.1.2. Öznitelik alt küme seçimi

Nitelik sıralama, öznitelik alt küme seçme algoritmalarının aksine birkaç özniteliği bulabilir. Büyük veri setlerinde, sadece en önemli ve tanımlayıcı özniteliklerin seçilmesi ve geri kalanının göz ardı edilmesi önemlidir [60-61]. Öznitelik alt kümesi seçimi üç şekilde gerçekleştirilebilir: Filtreleme, sarmalama ve tümleşik.

Tüm modellerin avantaj ve dezavantajları vardır. Genel olarak, filtreleme hızlıdır. Öğrenme modeli dâhil etmezler ve öznitelikleri (karşılıklı bilgi, veri tutarlılığı, vb.) seçmek veya atmak için eğitim verilerinin özelliklerini dikkate alırlar. Sarmalama ise özniteliklerin her alt kümesini değerlendirmek için bir öğrenme algoritması (bir sınıflandırıcı ya da kümeleme algoritması) içerir. Öğrenme algoritmasını dâhil ederek doğruluğun iyileştirmesini hedeflerler. Ancak, sarmalama hesaplama açısından maliyetlidir. Bu nedenle büyük veri setleri için uygulamaları kısıtlıdır. Tümleşik bir model sınıflandırıcı, eğitim sürecinde öznitelik seçimi yapar ve genellikle verilen öğrenme modellerine bağımlıdır. Genellikle sarmalama yaklaşımına göre daha hızlıdır. Aynı zamanda aşırı uyum olasılığı daha yüksektir [62]. Büyük eğitim seti olduğu durumlarda filtreleme kullanılabilir [58].

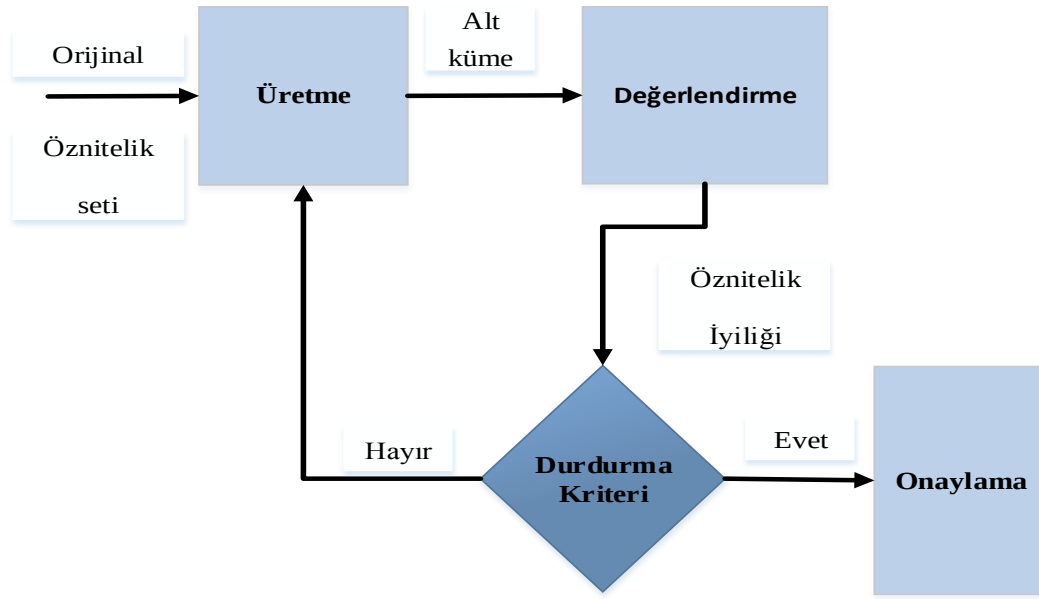
Öznitelik alt küme seçme yaklaşımının, bireysel öngörü gücüne göre nitelik sıralamasından daha iyi tahmin yeteneğine sahip olduğu inanılmaktadır. Daha önce belirtildiği gibi diğer nitelikler ile hesaba alındığında, tek başına tamamen yararsız olan tek bir nitelik dikkat çekici bir şekilde performansı artırabilir. Başka bir nitelik ile yüksek korelasyona sahip olan bir nitelik, belli bir alt kümede gereksiz olduğu için hiçbir ek fayda sağlayamaz. Öznitelik sıralama yaklaşımları bu senaryolar ile başa çıkmada yetersiz kalabilir [39].

İdeal olarak, öznitelik seçimi yöntemleri, öznitelik altkümesinde arama yapar ve bazı değerlendirme işlevine göre 2 aday alt küme arasında en iyisini bulmaya çalışır. Ancak bu işlem ayrıntılı ve en iyiyi bulmak için çalışır. Bu nedenle orta büyüklükte bir öznitelik seti boyutu (N) için, maliyet yüksektir ve pratik olmayabilir. Sezgisel veya rasgele arama yöntemleri hesaplama karmaşıklığını azaltmaya çalışır. Bu yöntemler son bulma için bir durdurma kriterine ihtiyaç duyar. Şekil 3.2’de görüldüğü gibi tipik bir öznitelik seçme yönteminin dört temel adımı vardır. Bunlar;

1. Bir sonraki aday alt kümesi oluşturmak için bir üretim prosedürü,
2. İnceleme altındaki alt kümeyi değerlendirmek için bir değerlendirme fonksiyonu,
3. Son bulma ya da durdurma kriteri ve
4. Alt kümenin geçerli olup olmadığını kontrol etmek için bir doğrulama işlemi.

Üretim prosedürü bir arama işlemidir [63-64]. Temel olarak, değerlendirme için özniteliklerin altkümelerini oluşturur. Üretim işlemi şu şekilde başlayabilir: (i) Hiçbir öznitelik olmadan. (ii) Tüm öznitelikler ya da (iii) rastgele özniteliklerin bir alt kısmı olduğunda. İlk iki durumda, öznitelikler tekrarlanarak eklenir veya kaldırılır, son durumda ise, öznitelikler ya tekrarlanarak eklenir veya kaldırılır veya daha sonra rastgele üretilir [55].

Bir değerlendirme fonksiyonu bazı üretim işlemi ile üretilen bir alt kümenin iyiliğini ölçer ve bu değer, daha önceki en iyi olan ile karşılaştırılır. Daha iyi olduğu tespit edilirse, o zaman daha önceki en iyi alt kümenin yerini alır. Uygun bir durdurma kriteri olmadan öznitelik seçme işlevi alt kümelerin uzayında ayrıntılı olarak ya da sonsuza kadar çalışabilir. Üretim işlemleri ve değerlendirme işlevleri bir durdurma kriter için seçimi etkileyebilir. Bir nesil prosedürüne dayalı durdurma kriterleri: (i) öznitelikleri önceden tanımlanan bir öznitelik sayısı seçilir ve (ii) önceden tanımlanmış bir yineleme sayısı ile belirlenir. Bir değerlendirme fonksiyonuna dayalı durdurma kriterleri: (i) herhangi bir özniteliğin eklenmesi (veya silinmesi) daha iyi bir alt küme vermezse ve (ii) değerlendirme fonksiyonuna göre uygun (optimal) bir alt küme elde edildiğinde son bulur. Bazı durdurma kriterleri tatmin edici sonuç elde edilene kadar çalışmaya devam edebilir. Öznitelik seçimi süreci bir doğrulama prosedürü için özniteliklerden seçilen bir alt küme çıkışı yaparak durur [64].



Şekil 3.2. Doğrulama ile öznitelik seçim süreci

4. ÖZNETELİK ALT KÜME SEÇİMİ

4.1. Filtreleme Yöntemi

Filtreleme yaklaşımı öznitelik seçme problemi için uzun süre popüler ve hesaplama açısından hızlı bir yaklaşım olarak kabul edilmektedir. Filtreleme yöntemleri, öğrenme algoritmasından bağımsız bir şekilde çalışırlar ve sadece verilerin içsel özelliklerine bakarak özniteliklerin arasındaki ilişkiyi değerlendirirler [17]. Filtreleme yöntemlerinin temel amacı önceki bazı bilgilerle göre en iyi öznitelikleri seçmektir. Filtreleme öznitelik seçimi yöntemleri iki kategoriden oluşur: nitelik skorlama ve alt küme arama yöntemleri. Nitelik skorlama yöntemi, her niteliğe bağımsız olarak bir ağırlık atar ve daha sonra hedef değişkeni ile ilgisine göre sıralar. Alt küme arama yöntemleri ise belirli bir değerlendirme ölçüsü kullanarak tüm öznitelik alt kümesini keşfederler.

Filtreleme teknikleri hızlı çalışır ancak sınıflandırıcı etkisini göz ardı eder. Her nitelik bireysel olarak incelenir. Böylece nitelik bağımlılıklarını yok sayıp, diğer öznitelik seçimi tekniklerine göre daha kötü sınıflandırma performansı elde edilebilir. Bu sorunu aşmak için, çok değişkenli filtreleme teknikleri tanımlanmıştır. Böylece, bir dereceye kadar nitelik bağımlılıklarının dâhil edilmesi amaçlanmıştır [65].

Tek değişkenli filtreleme yönteminde [66] her bir nitelik bireysel olarak incelenir. Bu yüzden nitelik bağımlılıkları yok sayıldığı için öznitelik seçme tekniklerinin diğer türlerine göre daha kötü sınıflandırma performansı gösterebilir.

Özellikle biyoinformatikte, mikrodizi verileri yüksek boyutlu olduğu için tek değişkenli filtreleme yöntemi hızlı ve verimli bir şekilde kullanılır. Bu alanda, tek değişkenli tekniklerin yaygınlığı egemendir. Genellikle DNA mikrodizi veri setlerinde öznitelik seçme tek değişkenli yöntem üzerine odaklıdır. Bunun nedeni şöyle açıklanabilir:

1. Tek değişkenli öznitelik sıralama tarafından sağlanan çıkış kolay anlaşılır,
2. Biyo-domain uzmanları laboratuvar teknikleri ile literatür aramaları keşfetmek için sonuç doğrulama yapmak istedikleri zaman, gen sıralama çıkışı, hedef ve beklentilerini

karşılaştırabilir. Uzmanlar gen etkileşimlerini dikkate aldıklarında seçim tekniklerine ihtiyaç hissetmezler.

3. Çok değişkenli gen seçimi teknikleri için ekstra hesaplama süresi gereklidir.

Aşağıda bazı tek değişkenli filtreleme yöntemlerinden bahsedilmiştir:

4.1.1. Tek değişkenli

Fisher ayırım oranı (FDR)

Fisher Ayırım Oranı (FDR) doğrusal öznitelik çıkarma yöntemi olarak yaygın bir şekilde kullanılır [39]. Bu yaklaşımın amacı, sınıfların merkezleri arasındaki mesafenin artırılması ve sınıf dağılımını minimize etmektir. Bu kavramı Eşitlik 4.1 ile gösterilebilir.

$$FDR = \sum_i^M \sum_{j \neq i}^M \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \quad (4.1)$$

Eşitlik 4.1'de, μ_i ve μ_j , i ve j sınıfın ortalamasıdır, sırayla σ_i^2 ve σ_j^2 , i ve j sınıfın varyansdır.

Relief

Relief, Kira ve Rendell tarafından önerilmiş örnek-tabanlı öznitelik filtreleme yöntemidir [65]. Bu yöntem, rasgele seçilen örneğe en yakın iki örneği bulmayı amaçlar. Bunlardan aynı sınıfta olan en yakın olan örnek seçilir ve Hit örnek (H) olarak adlandırılır. Farklı sınıftan en yakın kaybedilen örnek (M) (miss) olarak adlandırılır. İkinci olarak, örneklerin arasındaki özniteliklerin farkları hesaplanır. Algoritmanın sözde kodu Şekil 4.1'de gösterilmiştir.

Girdi: özniteliklerin ve sınıf değerlerin değeri ile eğitim örnekleri için bir vektör uzayı

Çıktı: Her bir özniteliğin W ağırlığı ile eğitim örnekleri için bir vektör uzayı

1. Tüm ağırlıkları $[A] = 0.0$ ayarlanır

2. $i = 1$ den m 'ye kadar, başla

2.1. Rastgele R_i örnek seç

2.2. En yakın hit H ve yakın miss M bul

2.3. $A = 1$ dan tüm öznitelikler için bu işlevi yap

2.3.1. $W(A) = W(A) - \text{diff}(A, R_i, H) / m + \text{diff}(A, R_i, M) / m$

...son

Şekil 4.1. Relief için sözde kod [66]

I_1 ve I_2 örneklerinde A özniteliğinin değerleri arasındaki farkları hesaplamak için $\text{diff}(A, I_1, I_2)$ kullanılır. Sonra, bunların değerlerine bağlı olarak tahmin kalitesi $W[A]$ hesaplanır.

4.2. Sarmalama Yöntemi

Sarmalama yöntemleri, nitelikler arası ilişkiyi de dikkate alarak, öznitelik altkütmesini öğrenme modeline bağlı olarak gerçekleştirmektedir. Sarmalama yöntemi, öznitelik alt kümelerini kestirim güçlerine dayalı olarak skorlamak için sınıflandırıcıdan bir kara kutu olarak faydalanırlar. Her bir yineleme adımında, özellikleri nesnel fonksiyondaki indirgeme miktarına dayalı olarak sıralanmaktadır. Daha sonra en alt sıradaki nitelikleri elemektedir. Çeşitli varyantlarda, geri yönlü nitelik eleme şeması ve doğrusal çekirdek kullanmaktadır [67].

Sarmalama yöntemleri, filtrelemeye göre hesaplama açısından maliyetlidir. Ancak genellikle daha iyi sonuçlar elde ederler. Bu nedenle bir indüksiyon algoritması ve eğitim verileri arasındaki spesifik etkileşimi ayarlarlar. Sarmalama, öznitelik alt kümesi arama ve model seçimi arasındaki etkileşimi içerir ve öznitelik bağımlılıklarını dikkate almaktadır. Sarmalama yöntemi iki sınıfa ayrılabilir: deterministik ve rastsal.

4.2.1. Deterministik

İleri öznitelik seçme yöntemi

İleri öznitelik seçme (İÖS) kullanışlı nitelikleri bulmak için basit bir arama stratejisidir. Bu algoritma boş bir nitelik alt kümesi ile başlar ve nitelikleri belirlenen sayıya ulaşana kadar veya daha iyi yaklaşım sonucu elde edilene kadar her adımda bir nitelik ekler. Bir adım için, her bir aday niteliği mevcut alt kümeye ekler ve ardından değerlendirir. En yüksek iyileşmeyi sağlayan nitelik, elde edilen alt kümede yer alır. Eğer en iyi yeni altkümeler bir eşikten daha fazla artıyorsa, algoritma bu altküme ile devam eder, aksi takdirde sonlanır.

Geri öznitelik seçme yöntemi

Geri öznitelik seçme (GÖS), olasılık tabanlı adım adım değişken seçme işlemidir. Bu yöntem, tüm modeli tahmin ederek başlar ve sonra kalan tüm giriş değişkenleri istatistiksel olarak önemli olana kadar, en kötü ortak değişkenleri birer birer kaldırır [69].

Geri arama, arama alanı keşfetmek için doğal ve iyi bilinen bir yöntemdir ama çalışma zamanı yüksek olabilir. Geriye doğru arama, eski değerlendirmelerin büyük karmaşıklığı nedeniyle yüksek boyutlu veri setleri için olanaksız olabilir [70]. İleri ve geriye doğru öznitelik seçme yöntemlerin bir dezavantajı da ilk kez bir nitelik seçildiğinde geri silinemez veya silindiğinde geri eklenemez. Bunun bir sonucu olarak, gereksiz nitelikler seçilebilir.

İki yönlü arama

Tek yönlü algoritmalarda, arama bir başlangıç düğümünden ilerler ve hedef düğüm karşılaşılan kadar devam eder. Çift yönlü arama önemli bir şekilde zaman tasarrufu sağlayabilir [71]. İki yönlü arama, potansiyel köklerden yapraklara ileri arama yaparken geriye doğru bu işlemi genişletir.

4.2.2. Rastsal

Genetik algoritma

Genetik algoritma (GA) tipik bir sarmalama yöntemidir. Holland tarafından ortaya atılan genetik algoritma aynı zamanda güçlü arama ve optimizasyon tekniğidir [26]. GA bir doğal seçim sürecini taklit eder. Çözüm uzayında yüksek olasılıklı en uygun çözümü bulmaya çalışır. GA, eğitim amacıyla verilen bir uzayın en verimli niteliklerini bulmada etkili bir yöntemdir.

GA tekrarlı bir şekilde bireylerden oluşan popülasyonunu günceller. GA genel olarak bir başlangıç popülasyon ile başlar, her tekrarda, bir uygunluk fonksiyonuna bağlı olarak bireyleri değerlendirir. Yeni popülasyonun, mevcut nesilden daha iyi olması garanti edilir. Bazı bireyler değişmeden yeni nesil geçerler. Diğer bireyler üzerinde mutasyon ve çaprazlama gibi genetik operatörler uygulanır ve böylece çocuklar oluşturulur. GA durdurma kriterine (iterasyon sayısı, uygun öznitelik alt kümesi vb.) ulaşana kadar bu işlemi bir kaç kez tekrar eder. Aşağıda GA için sözde kod verilmiştir (Şekil 4.2). n , popülasyondaki bireylerin sayısı; χ , her tekrarda çaprazlama ile yerleşen popülasyonun bir kısmı; ve μ mutasyon oranı.

Bireylerin temsili

Bir popülasyon, bireylerden oluşur. Popülasyondaki bireyler genellikle bit dizeleri ile temsil edilmektedir. Ardından, çaprazlama ve mutasyon kolay bir şekilde uygulanabilir. Ama bir bit dizisi olarak bireyi kodlayıp, sonra tekrardan çözmek için bazı yöntemlerin uygulanması gerekir.

Uygunluk fonksiyonu

GA, uygunluk fonksiyonu ile bireylerin uygunluğunu test eder.

```

Algoritma: GA ( $n, \chi, \mu$ )
//Nesil 0 e ilk deęerleri vermek:
 $k := 0$ ;
 $P_k$ :  $n$ 'nın bir popölasyonu rastgele-ölretilen bireylerin;
// $P_k$  deęerlendirilir:
Her  $i \in P_k$  için uygunluk ( $i$ ) hesaplanır;
yapmak
{ // Nesil oluřturmak  $k + 1$ :
// 1. Kopyala:
 $P_k$ 'nın  $(1-\chi) \times n$  üyeleri seęilir; çocuklar  $P_{k+1}$  e atılır;
// 2. Çaprazlama:
 $P_k$ 'nın  $\chi \times n$  üyeleri seęilir, çiftleřir, çocuklar ölretilir, çocuklar  $P_{k+1}$  e atılır;
// 3. Mutasyon:
 $P_{k+1}$ 'ın  $\mu \times n$  üyeleri seęilir; Her birinde rastgele seęilen bit ters olur;
// $P_{k+1}$  deęerlendirilir:
Her  $i \in P_k$  için uygunluk ( $i$ ) hesaplanır;
// Artım:
 $k := k + 1$ ;
}
 $P_k$ 'de en uygun bireyin uygunluk deęeri, yeterli deęildir; devam eder
 $P_k$  en uygun birey ise döner;

```

řekil 4.2. GA için sözde kod [72].

Genetik operatölrer

Yeni neslin bir kısmı χ , ise geri kalan çaprazlama ile oluřturulacaktır. Sonra $(1-\chi)$ bu nesilden, sonraki nesil e doğrudan kopyalanacaktır. Toplamda, $(1 - \chi) \times n$ birey kopyalanır.

Rulet tekerleđi

Rulet tekerleđi seřiminde, bireylerin seřilme olasılıđı, P (seřim = i) ařađıdaki gibi hesaplanır:

$$P(\text{seřim} = i) = \frac{Uygunluk(i)}{\sum_{j=1}^n Uygunluk(j)} \quad (4.2)$$

Bir rulet tekerleđi her bireyin uygunluk deđerine bađlı, muhtemel farklı boyutlardadır. Ařađıda sözde kodu verilmiřtir (řekil 4.3):

Algoritma: RULET TEKERLİĐİ SEŘİMİ ()

$r :=$ rastgele sayı, $0 \leq r < 1$;

$toplamlam := 0$;

her i birey için

{ $toplamlam := toplamlam + P$ (seřim = i);

Eđer $r < toplamlam$ ise

{ i e dön;

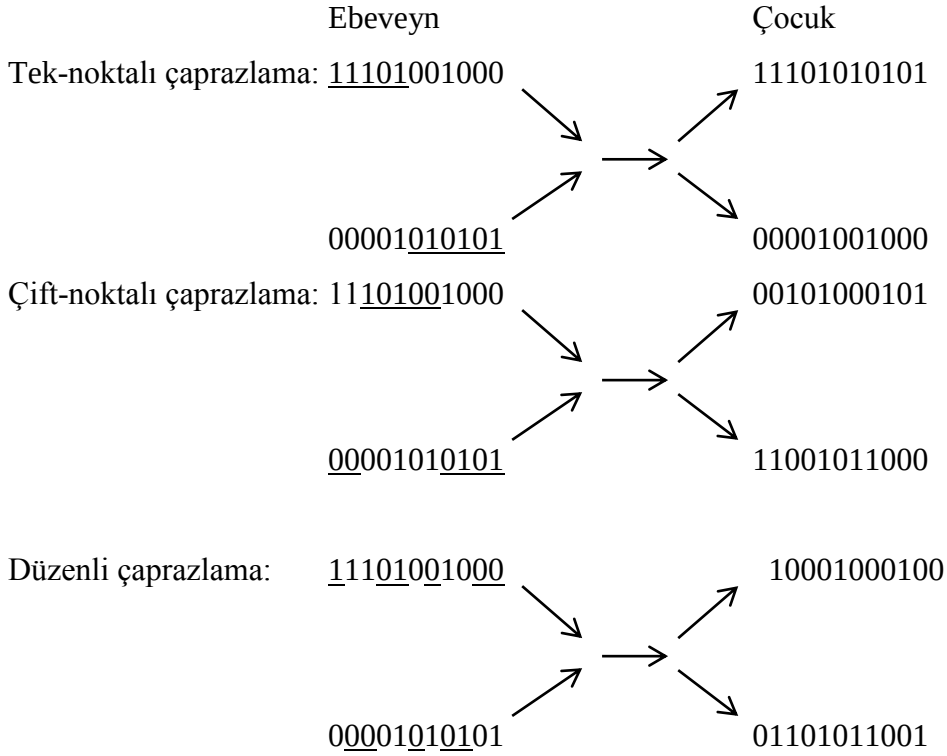
}

}

řekil 4.3. Rulet tekerliđi seřimi için sözde kod

Çaprazlama

Çaprazlama ile mevcut nesilden iki ebeveyn ile yeni çocuklar oluşturulabilir. Babanın bit dizisinden rastgele bir alt kısmı, annenin bit dizisinin rastgele bir alt kısmı ile yerleri deđiřtirilir. Ebeveynlerin $\frac{\chi \times n}{2}$ çiftleri oluşturmak için $\chi \times n$ bireye ihtiyaç var. Çaprazlama örnekleri ařađıda gösterilmiřtir.



Tek nokta, çaprazlama için bit dizisinde rastgele bir pozisyon belirler. İki-nokta çaprazlamada, iki pozisyon seçilir ve bölümlerin yerleri değiştirilir. Düzenli çaprazlamada, noktalar rastgele seçilir. Çocuklar ebeveynlerinden daha iyi veya daha kötü olabilirler. Aşağıda bir örnek için tek-noktalı çaprazlama gösterilmektedir.

1. Çaprazlama noktası seçin: Bit dizelerin uzunluğunu n varsayalım. 1 ve $n-1$ arasında rasgele bir sayı üretin. p adlandırın. Örnekte, bit dizinin uzunluğu 11 olursa, $p = 5$ olduğunu varsayalım.

2. Maskeleri oluştur: $mask_1$, $2^n - 2^p$ 'nin ikili gösterimi olacaktır; $mask_2, 2^p - 1$ 'nin ikili gösterimi olacak. Örnekte, $mask_1 2^{11} - 2^5 = 2016$, ikili formatı 11111100000 dır, $mask_2 2^5 - 1 = 30$, ikili formatı 11111 dur ve 11 bit olması için 00000011111.

3. Çocuk₁ üret: Çocuk₁=(ebeveyn₁ VE mask₁) YA (ebeveyn₂ VE mask₂). Örnek olarak:

ebeveyn₁ 11101001000 ebeveyn₂ 00001010101

mask₁ 11111100000 mask₂ 00000011111

VE 1110100000 VE 00000010101

 YA

=11101010101

4. Çocuk₂ üret: Çocuk₂=(ebeveyn₁ VE mask₂) YA (ebeveyn₂ VE mask₁), Örnek olarak:

ebeveyn ₁	11101001000	ebeveyn ₂	00001010101
mask ₂	00000011111	mask ₁	11111100000
VE	00000001000	VE	00001000000
YA			
=00001001000			

Mutasyon

Bu noktada, popülasyonun yeni nesli tamamlanmıştır: $(1 - \chi) \times n$ birey, kopyalanmıştır ve $\chi \times n$ birey bir çaprazlama operatörü tarafından üretilmiştir. Yeni nesilden μ 'nun bir belli oranı, rasgele olarak seçilir. Bu seçim, düzenli olasılık ile yapılır: bu işlem uygunluğa dayalı değildir. Her bir seçilen bireyde, bir bit rasgele seçilir ve değişir.

Mutasyon: 11101001000 $\Longrightarrow$ 11101011000

Mutasyon için örnek aşağıdaki gibi olabilir.

1. Mutasyon noktası seçin: bit dizeleri n uzunluğunda olduğunu varsayalım. Rasgele 0 den $n-1$ arasında bir sayı üretilir, p diyelim. Yukarıdaki örnekte, $p = 4$ varsayalım.
2. Bir maske oluşturun: maske 2^p 'nın ikili gösterimi olacaktır. Örneğin, maske $2^4=16$ olacaktır ki ikili de 00000010000 olacaktır.
3. Ters çevir: Yeni birey: = birey XOR maskesi. Örneğin:
Birey 11101001000
Maske 00000010000

XOR 11101011000

Mutasyon oranı düşük olmalıdır. Yani μ düşük bir değere sahip olacaktır. Mutasyon birey çeşitliliği açısından önemlidir [72].

4.3. Gömülü Yöntemi

Gömülü yöntemler, sınıflandırıcının eğitim fazına öznitelik seçimini entegre etmektedir. Bu nedenle, kullanılan öğrenme modeline göre değişiklik arz edebilir. Gömülü yöntemler, sarmalama yöntemine benzer. Öznitelik seçimi sınıflandırma ile bağlantılıdır. Bu bağlantı çok güçlüdür ki öznitelik seçimi sınıflandırıcı yapısına dâhil edilir.

Gömülü yöntemler, öznitelik seçimi ve sınıflandırma arasındaki etkileşim konusunda sarmalama yöntemler gibi aynı avantajları sunmaktadır. Buna ek olarak, daha iyi bir hesaplama karmaşıklığı sunarlar. Bu nedenle özniteliklerin seçimi doğrudan eğitim sürecinde sınıflandırıcı yapısına dâhil edilir. Mantıksal bağlaçlar tetikleme yöntemleri gömülü tür için örnek olabilir [16].

Çizelge 4.1. Arama modellerin avantajları, dezavantajları ve örnekler [16]

Arama modeli	Avantajlar	Dezavantajlar	Örnekler
Filtreleme	Tek değişkenli		
	Hızlı	Öznitelik bağımlılıkları	χ^2
	Ölçeklenebilir	göz ardı eder	Öklid uzaklık
	Sınıflandırıcıdan bağımsız	Sınıflandırıcı ile etkileşimi göz ardı eder	t-test Bilgi kazanç, Kazanç oranı
	Çok değişkenli		
	Öznitelik bağımlılıklarını modeller	Tek değişkenli tekniklerden daha	Korelasyon tabanlı öznitelik seçimi (CFS)
	Sınıflandırıcıdan bağımsızdır	yavaş	Markov blanket filtre (MBF)
	Sarma yöntemlere göre daha çok hesaplama karmaşıklığına sahip	Tek değişkenli tekniklerden daha az ölçeklenebilir	Hızlı korelasyon tabanlı öznitelik seçimi (FCBF)
		Sınıflandırıcı ile etkileşimi göz ardı eder	

Çizelge 4.1. (devam) Arama modellerin avantajları, dezavantajları ve örnekler [16]

Arama modeli	Avantajlar	Dezavantajlar	Örnekler
Sarmalama	Belirlenimci		
	Basit	Uygunluk (overfitting)	Sıralı ileri seçimi (İÖS)
	Sınıflandırıcı ile etkileşimi var	riski	Sıralı geriye ortadan kaldırılması
Randomize	Öznitelik bağımlılıklarını modeller	Randomize den daha eğilimli	Bean arama
	az hesaplama maliyeti var	Bir yerel optimuma da	
	Randomize yöntemlerden daha yoğun	algoritmalar saplanıyor (hırslı aram)	
Gömülü	Öznitelik bağımlılıklarını modeller	Sınıflayıcı bağımlı seçim	
	Sınıflandırıcı ile etkileşiminde yerel optimuma dan daha az eğilimli	yoğun hesaplama maliyetine sahip	Tavlama benzetimi (TB)
	Öznitelik bağımlılıklarını modeller	Sınıflayıcı bağımlı seçim deterministik algoritmaları göre daha çok uygunluk riski var	Rastgele mutasyon dagcılık Genetik algoritmalar Dağıtım algoritmaları Tahmini
Gömülü	Sınıflandırıcı ile etkileşimdedir	Sınıflandırıcı bağımlı seçimi	Karar ağaçlar
	İyi hesaplama maliyeti var		Ağırlıklı naive Bayes
	Sarma yöntemlerden daha karmaşıklık		DVM ağırlık vektörü kullanılarak öznitelik seçimi

5. SINIFLANDIRMA

Sınıflandırma yöntemleri, istatistik, makine öğrenmesi ve uzman sistemlerde yaygın bir şekilde kullanılmaktadır. Sınıflandırma algoritması, eğitim veri kümesini girdi olarak alır. Eğitim verisi ile sınıflandırma modeli oluşturulurken, test verisi sınama için gereklidir. Bu veriler, örneklerden ve her bir örnek de çeşitli nitelikler içerir. Nitelik değerleri iki grupta incelenebilir: Sıralı veya kategorik. Sınıf etiketi, genellikle kategorik niteliklerdendir. Sınıflandırma algoritması, test verisinde sınıf etiketini belirleme başarısı ile belirlenir.

Çeşitli sınıflandırma yöntemleri mevcuttur. Bunlardan bazıları: Karar ağacı, Yapay Sini Ağları, Bayes sınıflandırıcısı, Destek vektör makineleri ve k En yakın komşu olarak sayılabilir. Aynı tür uygulamada değişik sınıflandırma algoritmaları farklı sonuçlar verebilmektedir. Elde edilen öznitelik vektörlerine en uygun sınıflandırıcıyı seçebilmek için sınıflandırma yöntemlerinin özelliklerinin iyi bilinmesi gerekmektedir.

5.1. Naive Bayes

Naive Bayes sınıflandırıcı Bayes teoremine dayanır ve sınıflandırma performansı nispeten iyidir. Her bir veri örneği, n-boyutlu bir vektör $X = \{x_1, x_2, \dots, x_n\}$ ile temsil edilir, sırasıyla n öznitelik $A_1, A_2, \dots, A_n$ şeklinde tanımlanır. m adet sınıf $C_1, C_2, \dots, C_m$ ve X 'in de sınıfı bilinmeyen etiket olduğunu farz edelim. $P(C_i|X)$ Bayes teoremine göre aşağıdaki gibi ifade edilebilir [73].

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (5.3)$$

$P(X)$ tüm sınıflar için sabit olduğundan sadece $P(X|C_i)P(C_i)$ maksimum olması gerekmektedir. C_i Sınıfın önsel olasılığı bilinmiyorsa, genellikle, bu sınıfların olasılıkları eşit olduğu kabul edilir ki $P(C_1) = P(C_2) = \dots = P(C_m)$, bu yüzden sorun, $P(X|C_i)$ maksimize etmeğe dönüştürülür. C_i verildiğinde, $P(X|C_i)$, X verinin olasılığı olarak ifade edilir. $P(X|C_i)$ maksimize varsayımı ya da maksimum olasılığı denir. Aksi takdirde, $P(X|C_i)P(C_i)$ maksimize edilmesi gerekir.

Sınıfın önsel olasılığı $P(C_i) = \frac{S_i}{S}$ ile hesaplanabilir. S_i , C_i sınıfında S_i eğitim örneklerinin sayısıdır, S ise eğitim örneklerinin toplam sayısıdır.

Pek çok niteliğe sahip olan bir veri kümesinde, $P(X|C_i)$ hesaplama maliyeti çok olabilir. $P(X|C_i)$ hesaplanırken bunun için, sınıfların şartlı bağımsız olduğu kabul edilir. Örneğin sınıf etiketi göz önüne alındığında, varsayılan öznitelik değerleri, birbirinden bağımsız koşulludur. Öznitelikler arasında bağımlı bir ilişki yoktur. Buna göre;

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i) \quad (5.4)$$

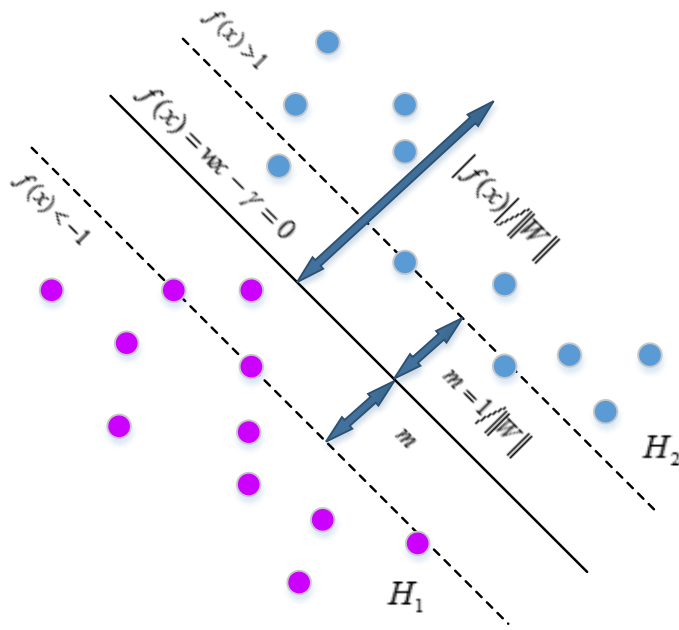
$P(x_1|C_i), P(x_2|C_i), \dots, P(x_n|C_i)$ olasılıkları, eğitim örnekten tahmin edilebilir. Bilinmeyen X örneğin sınıflandırması için her sınıf C_i için, $P(X|C_i)P(C_i)$ hesaplanır. X örneği Eşitlik 5.5.'e göre C_i kategorisine atanır.

$$P(C_i|X) \geq P(C_j|X), 1 \leq j \leq m, j \neq i \quad (5.5)$$

5.2. Destek Vektör Makinesi

Destek Vektör Makineleri (DVMs), bir ayırma hattı, düzlem veya alt düzlem oluşturarak, iki farklı sınıfı birbirinden ayırma prensibine dayanır. Bir Destek Vektör Makinesi kesinlikle bir makine değil, basit ve güçlü bir algoritmadır.

Bir sınıflandırma görevi genellikle bazı veri örneklerinden oluşan eğitim ve test verileri içerir. Eğitim setinde her örnek bir "hedef değer" (sınıf etiketi) ve birkaç nitelik (öznitelik)'den oluşur. DVM yönteminin amacı, sadece öznitelikleri verilmiş olan test setinde veri örneklerinin hedef değerini tahmin etmek için bir model oluşturmaktır [74]. DVM için doğrusal bir ikili sınıflandırma Şekil 5.1'deki gibi gösterilebilir. DVM sınıfı belirlemek için ayrımı maksimize eden bir alt düzlem ($w \cdot x = 0$) bulur. Alt düzlemle en yakın veri noktaları (destek vektörleri) ve alt düzlem arasındaki mesafeyi gösterir.



Şekil 5.1. Marjini maksimize eden ayrım düzlemi. (Mavi renkte olan 'o', olumlu bir veri noktası, yani $f('o') > 0$ ve mor renkte olan 'o' negatif veri noktasıdır, yani $f('o') < 0$) [75]

5.2.1. Doğrusal destek vektör makineleri

Ayrılabilir durum

Doğrusal makineler ayrılabilir veriler üzerinde eğitilmiştir. Eğitim verilerinin etiketini $\{x_i, y_i\}, i = 1, \dots, l, y_i \in \{-1, 1\}, x_i \in R^d$ varsayalım. Burada, negatif örnekleri pozitif örneklerden ayıran bazı alt düzlemler olduğunu varsayalım. Alt düzlem üzerinde yer alan X noktaları $w \cdot x + b = 0$ ile gösterilebilir ve w alt düzleme diktir. $b/\|W\|$ alt düzlemden dik mesafedir ve $\|W\|$, w 'nın Öklid normudur. d_+ (d_-) ayıran alt düzlemden en yakın pozitif (negatif) örneğe en kısa mesafe farz edelim. Bir ayırma alt düzlemin "kenarlık" i. $d_+ + d_-$ ile tarif edilir. Doğrusal ayrılabilir durumda, destek vektör algoritması basit bir şekilde büyük kenarlığa sahip olan bir ayırma düzlemi bulmaya çalışır. Bu kavram, aşağıdaki gibi gösterilebilir. Ve ayrıca tüm eğitim verilerin aşağıdaki kısıtlamaları karşıladıklarını varsayalım [75]:

$$x_i \cdot w + b \geq +1 \text{ for } y_i = +1 \quad (5.6)$$

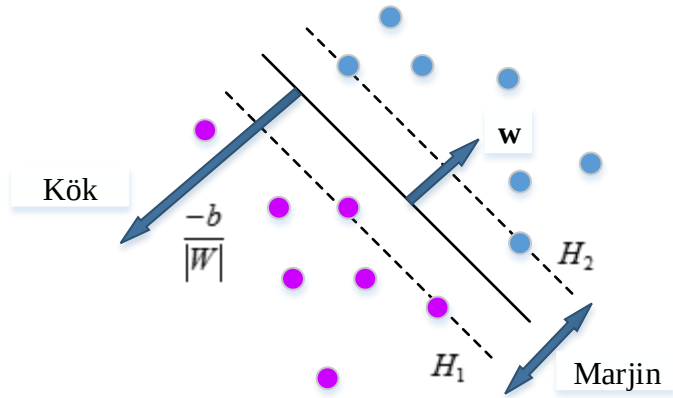
$$x_i \cdot w + b \leq -1 \text{ for } y_i = -1 \quad (5.7)$$

Bu eşitsizlikler bir dizi içine kombine edilebilir:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i \quad (5.8)$$

Eşitlik 5.6'de noktalar normal w ve kökten dikey uzaklık $|1 - b|/\|w\|$ ile $H_1 : x_i \cdot w + b = +1$ alt düzlemi üzerinde yer alırlar. Benzer bir şekilde, Eşitlik 5.7'de noktalar normal w ve kökten dikey uzaklık $|-1 - b|/\|w\|$ ile $H_2 : x_i \cdot w + b = -1$ alt düzlemi üzerinde yer alırlar. Sonuç olarak, $d_+ = d_- = 1/\|w\|$ ve marjin kolay bir şekilde $2/\|w\|$ tır.

H_1 ve H_2 paralellerdir ve aralarında hiçbir eğitim noktası düşmez. Böylece alt düzlemlerin çifti $\|w\|^2$ minimize ederek maksimum bir marjin elde eder. Böylece tipik bir iki boyutlu durum için Şekil 5.2'de gösterildiği gibi bir çözüm beklenir.



Şekil 5.2. Doğrusal ayrılabilir durum için alt düzlem ayrımı [75]

Ayrılmayan durum

Ayrılabilir veri için uygun olan yukarıdaki algoritma, ayrılmayan veri için uygulandığında, uygulanabilir çözüm bulamaz. Bu, amaç fonksiyonu ile giderilebilir. Aşağıda, Eşitlik 5.9 ve 5.10 kısıtlamalarını sadeleştirmek için ek olarak bir başka maliyet tanımlanır. Bu işlem, kısıtlamalarda pozitif gevşeklik değişkenleri ξ_i , $i=1, \dots, l$, $\xi_i \geq 0 \quad \forall i$, tarif ederek yapılabilir [76]:

$$x_i \cdot w + b \geq +1 - \xi_i \text{ for } y_i = +1 \quad (5.9)$$

$$x_i \cdot w + b \leq -1 + \xi_i \text{ for } y_i = -1 \quad (5.10)$$

Böylece, bir hatanın oluşması için, ilgili ξ_i , tekliği aşması gerekir. Böylece $\sum_i \xi_i$ eğitim hataların sayısına bağlı bir üst sınır elde edilir. Dolayısıyla hatalara ekstra bir maliyet atamak için doğal bir yol bulunur. Amaç fonksiyonu $\|w\|^2/2$ den $\|w\|^2/2 + C(\sum_i \xi_i)^k$ şeklinde minimize edilebilir. Burada C kullanıcı tarafından seçilen bir parametredir. Bu herhangi bir k pozitif tamsayı için bir dışbükey programlama problemi olarak; $k=2$ ve $k=1$ için ayrıca bir kuadratik programlama problemidir. $k=1$ olması ile birlikte ξ_i ve Lagrange çarpanları, Wolfe dual problemi şöyle maksimize edilir:

$$L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (5.11)$$

$$\sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C \quad (5.12)$$

Çözüm tekrardan verilebilir:

$$w = \sum_{i=1}^{N_S} \alpha_i y_i x_i \quad (5.13)$$

Burada, N_S destek vektörlerin sayısıdır. Bu durumda, uygun düzlem durumu ile tek farkı budur ki şimdi α_i , C den bir üst sınırı var. Özgün sorun (Primal problem) için Karush-Kuhn-Tucker koşulları gereklidir. Özgün Lagrange ise:

$$L_P = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} - \sum_i \mu_i \xi_i \quad (5.14)$$

μ_i Lagrange çarpanlarıdır ki ξ_i 'nin pozitifliğini güçlendirmek için tanıtlır. İlkel sorun için KKT şartları:

$$\frac{\partial L_P}{\partial \omega_v} = \omega_v - \sum_i \alpha_i y_i x_{iv} = 0 \quad (5.15)$$

$$\frac{\partial L_P}{\partial b} = - \sum_i \alpha_i y_i = 0 \quad (5.16)$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad (5.17)$$

$$y_i(x_i \cdot w + b) - 1 + \xi_i \geq 0 \quad (5.18)$$

$$\xi_i \geq 0 \quad (5.19)$$

$$\alpha_i \geq 0 \quad (5.20)$$

$$\mu_i \geq 0 \quad (5.21)$$

$$\alpha_i \{y_i(x_i \cdot w + b) - 1 + \xi_i\} = 0 \quad (5.22)$$

$$\mu_i \xi_i = 0 \quad (5.23)$$

Daha önce olduğu gibi, KKT tamamlayıcılık koşulları ve Eşitlik 5.22 ve 5.23 eşik b' i belirlemek için kullanabiliriz. Eşitlik 5.17 ve 5.18 ile birlikte göstermektedir ki $\xi_i = 0$ eğer $\alpha_i < C$. Böylece, b' i hesaplamak için eşitlik 5.22 kullanmak için $0 < \alpha_i < C$ herhangi bir eğitim noktası kullanabiliriz.

5.2.2. Doğrusal olmayan destek vektör makineleri

Karar fonksiyonu doğrusal bir fonksiyon olmayan durumlar için yukarıdaki yöntemler nasıl genelleştirilebilir? [77], eski bir hile (trick) [78], şaşırtıcı bir şekilde bunu yapmak için kullanılabileceğini göstermektedir.

İlk olarak, Φ adında bir eşleme kullanılarak, diğer bazı (muhtemelen sonsuz boyutlu) Öklid uzayına H , verileri eşleştirilmiş varsayalım:

$$\Phi: R^d \rightarrow H \quad (5.24)$$

Sonra eğitim algoritması sadece, $\Phi(x_i) \cdot \Phi(x_j)$ formun fonksiyonları üzerinde yani H , nokta ürünleri yoluyla verilere bağlı olacaktır. Eğer bir "kernel fonksiyonu" K olsaydı, öyle ki, $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$, sadece eğitim algoritmasında K kullanmanız gerekir ve hatta Φ ne olduğunu bilmek gerekmiyor. Bunun bir örneği:

$$K(x_i, x_j) = e^{-\|x_i - x_j\|^2 / 2\alpha^2} \quad (5.25)$$

Bu özel örnekte, H sonsuz boyutludur, bu yüzden Φ ile çalışmak çok kolay değildir. Ancak, eğitim algoritmasında her yerde eğer $x_i \cdot x_j$ yerine $K(x_i, x_j)$ olursa, algoritma sonsuz boyutlu uzayda olan bir destek vektör makinesi üretecektir. Daha önce anlatılan tüm hususlar göz önünde tutulur, ancak doğrusal bir ayrım farklı bir uzayda yapılmaktadır. Bunun için, w gereklidir ki H da yani (Eşitlik. (5.26)) da olacaktır. Ancak, test fazında, w ile verilen bir x test noktasının nokta ürünlerinin hesaplanması ile bir DVM kullanılır, daha spesifik olarak aşağıdaki şekilde hesaplanır.

$$f(x) = \sum_{i=1}^{N_s} \alpha_i y_i \Phi(s_i) \cdot \Phi(x) + b = \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x) + b \quad (5.26)$$

s_i destek vektörleridir. Yani açıkça $\Phi(x)$ hesaplamayı önleyebiliriz ve yerine $K(s_i, x) = \Phi(s_i) \cdot \Phi(x)$ kullanabiliriz.

Örüntü tanıma problemi için aşağıdaki kerneller örnek olabilir:

$$K(x, y) = (x \cdot y + 1)^p \quad (5.27)$$

$$K(x, y) = e^{-\|x - y\|^2 / 2\alpha^2} \quad (5.28)$$

$$K(x, y) = \tanh(kx \cdot y - \delta) \quad (5.29)$$

Eşitlik 5.27 bir sınıflandırıcı sonuçlar ki p derecesinde birçok terimlidir; Eşitlik (5.28), bir gauss radyal temel fonksiyonu (RTF) sınıflandırıcı elde eder ve Eşitlik 5.29, iki katmanlı

sigmoidal sinir ağıının belirli bir türünü elde eder.

RTF durumunda, merkezlerinin sayısı (N_s Eşitlik 5.26'de), merkezlerin kendileri (s_i), ağırlıklar (α_i), ve eşik (b) hepsi otomatik olarak DVM eğitimi tarafından oluşturulur ve Gauss RTFler [79] durumu için klasik RTFler karşılaştırdığında mükemmel sonuçlar elde eder. Sinir ağı durumunda, birinci tabaka, ağırlıkların N_s kümelerinden oluşur, her küme d_L ağırlıklarından oluşur (veri boyutu), ve ikinci tabaka N_s ağırlıklarından (α_i), oluşur, bu yüzden bir değerlendirme, sadece sigmoidlerin bir ağırlık toplamının almasını gerektirir, böylece kendileri destek vektörleri ile test verilerin nokta ürünleri üzerinde değerlendirilir. Böylece sinir ağı durum için, bu düzenlemenin yapısı (ağırlıkların sayısı), DVM eğitim ile belirlenir. Hiperbolik tanjant çekirdek sadece κ ve δ parametrelerin belli değerleri için Mercer şartını sağlar [76]. Son olarak, yukarıda tarif edilen DVM sınıflandırıcılar ikili sınıflandırıcılar olmasına rağmen, kolaylıkla çok sınıflı durumlar genişletilebilirler.

5.3. Karar Ağacı

Karar ağacı (KA), öğrenen bir ağaç yapısıdır ki her yaprak olmayan düğüm, bir özelliğin üzerinde bir testi temsil eder, her dal da testin bir sonucunu temsil etmektedir. Her yaprak düğüm ise sınıf etiketini temsil eder. Sınıflandırma sonucunda elde edilen yapı kolay anlaşıldığından karar ağaçları popüler olmuştur. Karar ağaçları oluşturmak için farklı yöntemler mevcuttur, öznitelik değerleri ve bir sınıf etiketi arasındaki ilişkiyi temsil eden her dal ile bir ağaç yapısında verilen eğitim verileri özetlenir. Bu işlev, sınıfların ayırımında nitelik değerlerin potansiyeli üzerinden testlere göre ayarlanmış eğitim verisini ayırmakta yinelemeli olarak çalışır. Bu işlem karar ağacı olan çekirdeği ID3 adındaki orijinal versiyona dayanmaktadır.

ID3 (İterative Dichotomiser) yöntemi, bilgi teorisini kullanarak karar ağacını oluşturur [80]. Bu yöntem bir veri kümesinden yüksek bilgi kazanç değerine sahip olan ayırmacı öznitelikleri seçer. Bir öznitelik değeri ile ilişkili bilgilerin miktarı oluşum olasılığına bağlıdır.

Bilgi ölçmek için kullanılan kavram, entropinin denir ki her hangi bir veri kümesinin rastgele miktarını ölçmek için kullanılır. Bir kümedeki tüm veriler tek bir sınıfa ait olduğunda, hiçbir belirsizlik olmadığı ve entropinin sıfır olduğu anlamına gelir. KA

sınıflandırma yöntemin amacı, tekrarlı bir şekilde her son alt kümede ki tüm örnekleri tek sınıfa ait olana kadar alt küme verilerin veri kümesini parçalamaktır. Eşitlik 5.30, entropi hesaplamasının denklemini göstermektedir. Entropi değeri 0 ve 1 arasında değişir ve olasılıkların hepsi aynı olduğunda maksimumu elde eder. Verilen olasılıklar $p_1, p_2, \dots, p_s$ ki $\sum_{i=1}^s p_i = 1$,

$$\text{Entropi: } H(p_1, p_2, \dots, p_s) = - \sum_{i=1}^s (p_i \log(\frac{1}{p_i})) \quad (5.30)$$

Bir D veri kümesi verildiğinde, $H(D)$ veri kümenin bir alt kümesinin miktarını bulur.

Alt küme, s adet yeni alt kümeye bölündüğünde $S = \{D_1, D_2, \dots, D_s\}$ yine de bu altkümelerin entropisine bakabiliriz, tüm örnekleri aynı sınıfta ise veri kümesinin bir alt kümesi tamamen düzenlenmiştir. ID3, yüksek bilgi kazanç elde eden özneliği, ayırmacı öznelik olarak seçer. ID3 algoritması, Eşitlik 5.31 ile bilgi kazancı hesaplar.

$$\text{Kazanç } (D, S) = H(D) - \sum_{i=1}^s p(D_i) H(D_i) \quad (5.31)$$

C4.5, kazanç oranının aracılığıyla ID3 'ün bir devamıdır [81]. Ayırım amaçla, C4.5 en büyük kazanç oranını kullanır ki bilgi kazancın ortalamasından daha büyük olduğunu sağlar.

$$\text{Kazanç oranı } (D, S) = \frac{\text{Kazanç}(D, S)}{H(\frac{|D_1|}{|D|}, \dots, \frac{|D_s|}{|D|})} \quad (5.32)$$

C5.0 algoritması, iyileştirme kullanarak üretilen ağaçların performansını artırır ki birçok gürültü veriden oluşan eğitim verisinde yardımcı olmuyor. CART (sınıflandırma ve regresyon ağaçları karar verme amacıyla bir ikili ağacı üreten bir süreçtir [82]. CART eksik verileri işleyen ve bir budama strateji içerir. KAÖPİ (Karar Ağaçları Ölçeklenebilir Paralleleştirilebilir İndüksiyon) algoritması en iyi bölünmeyi bulmak için gini endeksi olarak adlandırılan bir saf olmama fonksiyonu kullanır [83]. Eşitlik 5.33, bir veri kümesi için gini tanımlar:

$$\text{gini}(D) = 1 - \sum p_j^2 \quad (5.33)$$

D de, p_j , C_j sınıfın frekansı olarak tanımlanır, D_1 ve D_2 alt kümesinde D nin ayrımcılık iyiliği aşağıda ki gibi tanımlanır:

$$\text{gini}_{\text{split}}(D) = \frac{n_1}{n(\text{gini}(D_1))} + \frac{n_2}{n(\text{gini}(D_2))} \quad (5.34)$$

5.4. K En Yakın Komşu (KEYK)

KEYK sınıflandırıcı, eğitim kümesinin sınıf etiketlerine dayanarak bilinmeyen sınıf etiketini bulmayı hedefler. Bu veri kümesi büyük bir matristen oluşabilir. Satırlar eğitim vektörleri (N) ve sütunlar eğitim örnekleri (M) olarak adlandırılır. Sınıf etiketini temsil eden ek bir sütun daha vardır. Sınıflandırıcı bu şekilde çalışır ki ilk olarak, belirli bir mesafe ölçüsü (örneğin, Öklid ya da Manhattan) kullanarak sorgu vektörü ve tüm eğitim vektörlerin arasındaki mesafeleri hesaplar. Eğer bir sorgu M boyutlar ile $X = \{X_1, X_2, \dots, X_M\}$ olarak tanımlanırsa ve N eğitim vektörlerin her biri $Y_{NM} = \{Y_{N1}, Y_{N2}, \dots, Y_{NM}\}$, ve $L = \{L_1, L_2, \dots, L_N\}$, eğitim vektörlerin sınıf etiketi olursa, Öklid ölçüsü kullanarak bir eğitim vektörü Y ve bir sorgu X arasındaki mesafe şöyle hesaplanır:

$$D(X, Y) = \sqrt{\sum_{i=1}^M (X_i - Y_{Ni})^2} \quad (5.36)$$

Burada $D(X, Y)$, iki vektör X ve Y in mesafedir. Öklid mesafesi yaygın olarak, birçok veri madenciliği uygulamalar da mesafeleri hesaplamak için kullanılır, başka bir metrik Manhattan mesafesi dır ki Öklid mesafesine göre sade ve daha düşük maliyete sahiptir [84-85]. Manhattan mesafesi Eşitlik 5.37'de ifade edilir:

$$D(X, Y) = \sum_{i=1}^M (|X_i - Y_{Ni}|) \quad (5.37)$$

Mesafe hesaplama fazının sonunda, bir sorgu ve N eğitim vektörlerin her birinin bütün örneklerinin arasındaki toplanmış mesafelere karşılık gelen N mesafeler karşılaştırılır. Sonra sınıflandırıcı, K-minimum mesafeler elde eder ve K en yakın komşular veya KEYK olarak bilinen K en yakın eğitim vektörleri ile ilişkili sınıf etiketleri ile birlikte azalan bir şekilde sıralar. Son olarak, sınıflandırıcı en çok karşılaşılan KEYK ye sorguyu atamak için

KEYK üzerinde bir oylama gerçekleştirir. Ancak, her sorgu için tekrarlanan ilaveleri içerdiğinden dolayı mesafe hesaplama çok zaman alıcı bir işlemdir. Böylece, KEYK kısmını hızlandırmak için mesafe hesaplama bölümü paralel yapılabilir [86].

5.5. Doğrusal Ayırıcı Analizi

Doğrusal Ayırıcı Analizi (DAA) bir genel öz-değer çözer. C sınıfları ve n sayısı kadar d – boyutlu eğitim örneklerin olduğunu ve n_c , c sınıfın eğitim örneklerinin sayısını gösterdiğini varsayalım. 1 uygun uzunlukta hepsi-bir vektör olduğunu göstermektedir.

S_w sınıflar içi toplam kare matrisi, S_b sınıflar arasındaki toplam kare matrisi ve S_t toplam kare matrisidir.

$$S_w = \frac{1}{n} \sum_{c=1}^C (\hat{X}_c - m_c 1^T)(\hat{X}_c - m_c 1^T)^T \quad (5.38)$$

$$S_b = \frac{1}{n} \sum_{c=1}^C n_c(m_c - m)(m_c - m)^T \quad (5.39)$$

$$S_t = \frac{1}{n} (\hat{X} - m 1^T)(\hat{X} - m 1^T)^T \quad (5.40)$$

$\hat{X} \in \mathbb{R}^{d \times n}$ Bir veri matrisidir ki sütunlar eğitim örnekleridir, $\hat{X}_c$, C sınıfına ait eğitim örneklerin veri matrisidir, m , tüm eğitim örneklerin ortalama vektörüdür, m_c , C sınıfına ait eğitim örneklerin ortalama vektörü ve T matris devriğidir.

DAA, bir lineer dönüşüm matrisi $W \in \mathbb{R}^{d \times (C-1)}$ hesaplar ve genelde $d \gg C$. Dönüşüm matrisi, düşük boyutlu bir uzaya orijinal yüksek boyutlu uzaydan verileri atarak, sınıf içi mesafeyi en aza indirirken sınıflar arası mesafeyi maksimize eder. Geleneksel DAA, optimizasyon problemini çözerek uygun dönüşüm matrisi W_{DAA} bulur.

$$W_{DAA} = \underset{W}{\operatorname{argmaxtrace}} (W^T S_b W (W^T S_t W)^{-1}) \quad (5.41)$$

Eşitlik 5.41'de, S_t toplam kare matrisi tekil olmadığı süreçte, W_{DAA} çözüm matrisi, sıfır olmayan öz değerlere ait $(S_t^{-1}S_b)$ matrisin üst öz-vektörlerden oluşur.

Toplam saçılım matrisi S_t , tam bir rütbesi olmadığında, W_{DAA} , sıfır olmayan öz değerlere ait $(S_t^+S_b)$ öz değerlerinden oluşur ki S_t^+ , S_t sözde tersini (pseudo-inverse) ifade etmektedir [87].

5.6. Sınıflandırıcı Performansı

Burada, sadece iki sınıflı olan sınıflandırma problemleri ele alınmıştır. Her örnek I , pozitif ve negatif sınıf etiket kümesinin $\{p, n\}$ bir elemanına eşleştirilmiştir. Bir sınıflandırma modeli (veya sınıflandırıcı) örneklerden, öngörülen sınıflara bir eşleşmedir. Bazı sınıflandırma modelleri devamlı bir çıktı üretmektedir (örneğin, bir örneğin sınıf üyelik olasılığın tahmini) ki sınıf üyeliğini tahmin etmek için farklı eşikler uygulanabilir. Diğer modeller, örneğinin sadece öngörülen sınıfını gösteren bir ayrık sınıf etiketi üretir. Doğru ve tahmin sınıfları teşhis etmek için bir model tarafından üretilen sınıf tahminleri için $\{Y, N\}$ etiketleri kullanılır.

Bir sınıflandırıcı ve örneği göz önüne alındığında, dört olası sonuç vardır. Eğer örnek pozitif ise ve pozitif olarak sınıflanmış ise bir doğru pozitif olarak sayılır; eğer negatif olarak sınıflanmış ise, yanlış negatif olarak sayılır. Eğer örnek negatif ise ve negatif olarak sınıflanmış ise doğru negatif olarak sayılır, Eğer pozitif olarak sınıflanmış ise yanlış pozitif olarak sayılır.

Bir sınıflandırıcı ve örneklerin kümesi göz önüne alındığında, bir iki-iki hata matrisi, örneklerin kümesinin durumlarının temsili için inşa edilebilir. Bu matris, birçok ortak ölçütlerin temelini oluşturur.

Şekil 5.8 bir hata matrisi ve birkaç ortak ölçümlerin denklemlerini gösterir. Büyük diyagonal boyunca olan numaralar yapılan doğru kararları temsil eder. Bir sınıflandırıcının gerçek pozitif oranı (ayrıca isabet oranı ve geri çağırma denir) aşağıdaki gibi tahmin edilmektedir [88].

$$\text{Sınıflandırma doğruluğu} = \frac{DP + DN}{DP + YP + DN + YN} \quad (5.42)$$

$$\text{Hata oranı} = \frac{YP + YN}{DP + YP + DN + YN} \quad (5.43)$$

Sınıflandırıcının doğru pozitif oranı:

$$\text{DP oranı} \approx \frac{\text{Doğru sınıflanan pozitifler}}{\text{Tüm pozitifler}} \quad (5.44)$$

Sınıflandırıcının yanlış pozitif oranı:

$$\text{YP oranı} \approx \frac{\text{Yanlış sınıflanan negatifler}}{\text{Tüm negatifler}} \quad (5.45)$$

Aşağıda hata matrisi (Şekil 5.3) ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir.

		Doğru sınıf	
		p	n
Tahmin edilen sınıf	P	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	N	Yanlış Negetif (YN)	Doğru Negetif (DN)

Şekil 5.3. Hata matrisi

$$\text{YP oranı} = \frac{YP}{N} \quad (5.46)$$

$$\text{DP oranı} = \frac{DP}{P} \quad (5.47)$$

$$\text{Hassasiyet} = \frac{DP}{DP + YP} \quad (5.48)$$

$$\text{Yeniden çağırma} = \frac{DP}{P} \quad (5.49)$$

$$\text{Duyarlılık} = \text{yeniden çağırma} \quad (5.50)$$

$$\text{Seçicilik} = \frac{\text{Doğru negatifler}}{\text{yanlış pozitifler} + \text{Doğru negatifler}} \quad (5.51)$$

$$\text{Seçicilik} = 1 - \text{yp oranı} \quad (5.52)$$

$$\text{Pozitif tahmin değeri} = \text{hassasiyet} \quad (5.53)$$

$$\text{Doğruluk} = \frac{DP + DN}{P + N} \quad (5.54)$$

$$\text{F - ölçü} = \frac{2}{1/\text{hassasiyet} + 1/\text{yeniden çağırma}} \quad (5.55)$$

6. GERÇEKLEŞTİRİLEN ÇALIŞMA

6.1. Kullanılan Veri Kümeler

Önerilen model UCI'dan alınan dört farklı veri kümesinde denenmiştir. Bu veri kümelerine ait örnek sayısı, nitelik ve içerdikleri sınıflar Çizelge 6.1'de gösterilmiştir.

Çizelge 6.1. UCI veri tabanından alınan veri kümeleri

Veri kümesi	Nitelik sayısı	Örnek sayısı	Sınıf sayısı
WDBC	32	569	2
SPECTF	45	267	2
WPBC	35	198	2
Statlog	14	270	2

6.1.1. Meme kanseri veri kümesi (WDBC)

WDBC veri kümesi, 357 iyi huylu, 212 kötü huylu olmak üzere 569 örnek içermektedir. Her örnek 32 nitelikten oluşmaktadır. İlk iki tanesi benzersiz tanımlama numarası ve tanı durumunu göstermektedir (kimliği, tanı (iyi huylu / kötü huylu), 30 gerçek değerli giriş nitelikleri). Kalan 30 orijinal nitelik seti olarak kullanılır. Şekil 6.1, WDBC niteliklerinin açıklamasını göstermektedir [12].

1) Kimlik numarası
2) Tanı (M = kötü huylu, B = iyi huylu)
3-32) On gerçek değerli öznitelikler her hücre çekirdeği için hesaplanır:
a) Radius (çevre üzerindeki noktalardan merkeze mesafelerin ortalaması)
b) Doku (gri ölçekli değerlerinin standart sapması)
c) Çevre
d) Alan
e) Yumuşaklık (yarıçap uzunluklarda yerel varyasyon)
f) Özlülük ($\text{çevre}^2 / \text{alan} - 1.0$)
g) Konkavlık (Konturun konkav kısımlarının şiddeti)
h) Konkav noktalar (Konturun konkav kısımlarının sayısı)
i) Simetri
j) Fraktal boyut ("kıyı şeridi yaklaşım" -1)

Şekil 6.1. WDBC veri kümesi nitelik ve açıklamaları

6.1.2. SPECTF kalp veri kümesi

SPECTF veri kümesi, görüntülerin teşhisini açıklamaktadır. Veri, iki kategoride sınıflandırılır: Normal ve anormal. Veri, 55 normal ve 212 anormal sınıfa ait toplam 267 örnek içermektedir. Her hasta için 45 (1 ikili sınıf + 44 sürekli) nitelik modelinden oluşmaktadır. Bu deneyde, eğitim kümesi 187 (70%) örneğe sahip ve test kümesi 80 (30%) örnekten oluşmaktadır [12].

6.1.3. Meme kanseri wisconsin (Prognostik) veri kümesi (WPBC)

WPBC veri kümesi, 47 hastalığı tekrarlama (nüksetme) örneği ve 151 hastalığı tekrarlama (nüksetmeme) örneği olmak üzere 198 örnek içerir. Her örnek 35 nitelikten oluşmaktadır. İlk iki tanesi benzersiz kimlik numarasını ve tanı durumunu göstermektedir. Bu verinin, 136 (%70) eğitim kümesi ve 58 (%30) test kümesi olarak kullanılmıştır. Şekil 6.2, WPBC niteliklerinin açıklamasını göstermektedir [12].

- 1) Kimlik numarası
- 2) Tanı (R = nüksetme (recurrent), N = nüksetmeme (nonrecurrent))
- 3) Zaman (tekrarlama zamanı eğer öznitelik 2 (sınıf) = R, hastalık-serbest zamanı eğer öznitelik 2 (sınıf) = N)
- 4-33) Her hücre çekirdeği için on gerçek-değerli öznitelik hesaplanır:
 - a) Yarıçap (çevre üzerindeki noktalara merkezden mesafelerin ortalaması)
 - b) Doku (gri-ölçekli değerlerinin standart sapması)
 - c) Çevre
 - d) Alan
 - e) Düzgünlüğü (yarıçap uzunluklarında yerel varyasyon)
 - f) Kompakt ($\text{çevre}^2 / \text{alan} - 1.0$)
 - g) Konkavite (kontur konkav noktaların şiddeti)
(severity of concave portions of the contour)
 - h) konkav noktaları (kontur konkav noktaların sayısı)
 - i) Simetri
 - j) Fraktal boyut ("kıyı şeridi yaklaşımı" - 1)

Şekil 6.2. WPBC veri kümesi nitelik ve açıklamaları

6.1.4. Statlog (Kalp) veri kümesi

Statlog (kalp) veri kümesi, 270 örnek içerir 150 örnek olumsuz sınıfa ve kalan 120 örnek olumlu sınıfa aittir. Her örnek 14 nitelikten oluşmaktadır. Tanı durumu ve 13 gerçek-değer girdi nitelikleri göstermektedir. Bu verinin, 189 (70%) eğitim kümesi ve 81 (30%) test kümesi olarak kullanılmıştır. Şekil 6.3, veri kümesinin nitelikleri açıklanmıştır [12].

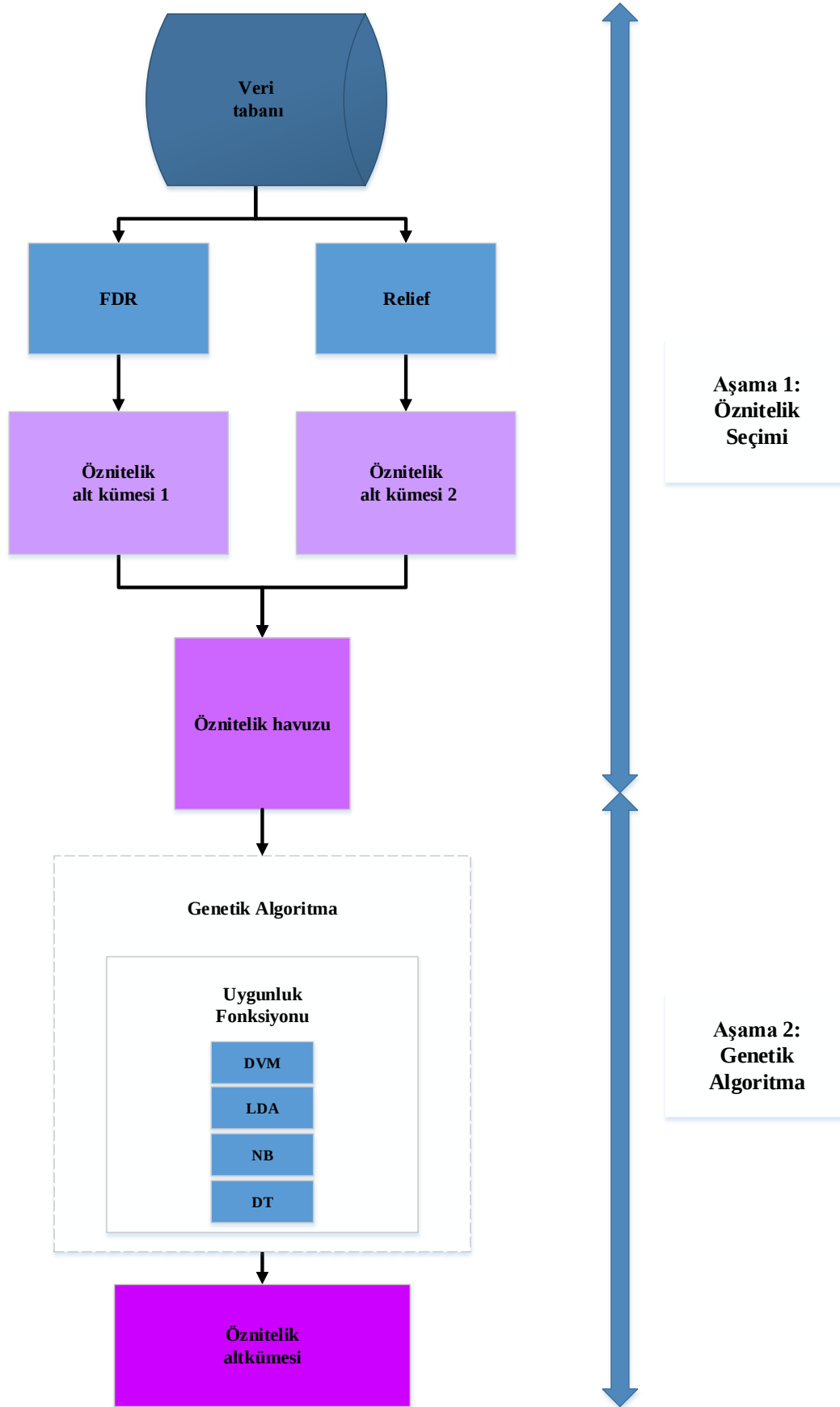
1. Yaş
2. Cinsiyet
3. Göğüs ağrısı tipi (4 değerli)
4. İstirahat kan basıncı
5. Mg/dl'de serum cholestoral
6. Açlık kan şekeri > 120 mg/dl
7. Dinlenme elektrokardiyografi sonuçları (0,1,2 değerleri)
8. Elde edilen maksimum kalp hızı
9. Egzersiz ki bağlı angına
10. Eski zirve = ST depresyon ki dinlemeye bağlı egzersiz tarafından kaynaklanan
11. ST segmentin zirve egzersiz eğimi
12. Flourosopy tarafından renklenen büyük damarların sayısı (0-3)
13. Talasemi: 3 = normal; 6 = sabit kusur; 7 = tersinir kusur

Şekil 6.3. Satlog veri kümesi nitelik ve açıklamaları

6.2. Deneysel Çalışma

Bu bölümde önerilen model, UCI'dan alınan dört farklı veri kümesinde denenmiştir. Her bir veri kümesi için belirlenen öznitelikler ve bunlara bağlı elde edilen sınıflandırma başarıları raporlanmıştır. Önerilen model iki aşamadan oluşmaktadır. İlk aşama, öznitelik havuzunun oluşturulması aşamasıdır. Burada iki filtreleme yöntemi kullanılmıştır. İkinci aşama, GA ile özniteliklerin belirlenmesi aşamasıdır. GA, etkin bir uygunluk fonksiyonu ile popülasyonları değerlendirir, yeni bir popülasyon oluşturmak için çaprazlama ve mutasyon işlemlerini uygular.

Arama sürecinin sonunda, GA en yüksek uygunluk değeri (sınıflandırma doğruluğu) ve öznitelik alt küme boyutu küçük olan en uygun öznitelik alt kümeyi seçer. Bu aşamada, GA her bir sınıflandırma yöntemi için ayrı ayrı çalışır. Her bir durum ayrı öznitelik alt kümesini oluşturur (Şekil 6.4).



Şekil 6.4. Genetik algoritma tabanlı etkin öznitelik seçimi

6.2.1. Birinci aşama: Öznitelik havuzu oluşturulması

Bu aşamada, FDR ve Relief ile orijinal nitelik kümesine uygulanır. Her filtreleme yöntemi bir öznitelik alt kümesi oluşturur ve sonuçta iki öznitelik altkümesi elde edilir.

WDBC

Çizelge 6.2’de tüm nitelikler için FDR değerleri gösterilmiştir. Çizelge 6.3’de 30’uncü nitelik 3,3916 FDR değeri ile en yüksek değere ve 21’nci nitelik 0,0001 değeri ile en düşük değere sahiptir. Çizelge 6.6’dan gösterdiği gibi en yüksek FDR değerine sahip olan 9 öznitelik {30, 25, 10, 23, 5, 3, 26, 9, 6} seçilmektedir. Bir özniteliğin FDR değerinin yüksek olması, öznitelik sınıflar arası mesafeyi maksimum etmede ve aynı zamanda sınıf dağılımını minimize etmekte başarılıdır. Bu nedenle, FDR yöntemi daha iyi ayırma yeteneğine sahip olan öznitelikleri seçerek daha iyi sonuçlara ulaşmakta yardımcı olabilir.

Çizelge 6.2. WDBC veri kümesi nitelikleri için FDR değeri

Nitelik	3	4	5	6	7	8	9
FDR	2,1036	0,4502	2,2536	1,7328	0,3196	1,0456	1,7511
Nitelik	10	11	12	13	14	15	16
FDR	2,7030	0,2542	0,0003	0,8019	0,0002	0,7507	0,6912
Nitelik	17	18	19	20	21	22	23
FDR	0,0098	0,1942	0,1616	0,4295	0,0001	0,0142	2,6999
Nitelik	24	25	26	27	28	29	30
FDR	0,5639	2,8129	1,9396	0,4500	0,9840	1,5360	3,3916
Nitelik	31	32					
FDR	0,3869	0,2231					

Çizelge 6.3. WDBC veri kümesinin niteliklerinin sıralanmış FDR değerleri

Nitelik	30	25	10	23	5	3	26
FDR	3,3916	2,8129	2,7030	2,6999	2,2536	2,1036	1,9396
Nitelik	9	6	29	8	28	13	15
FDR	1,7511	1,7328	1,5360	1,0456	0,9840	0,8019	0,7507
Nitelik	16	24	4	27	20	31	7
FDR	0,6912	0,5639	0,4502	0,4500	0,4295	0,3869	0,3196
Nitelik	11	32	18	19	22	17	12
FDR	0,2542	0,2231	0,1942	0,1616	0,0142	0,0098	0,0003
Nitelik	14	21					
FDR	0,0002	0,0001					

Çizelge 6.4’de tüm nitelik ve Relief değerleri görülmektedir. Çizelge 6.5, öznitelik ağırlık değerlerinin azalan bir şekilde sıralanmış halini göstermektedir. 24’üncü nitelik 0,0207 değeri ile en yüksek değere ve 12’nci nitelik -0,0137 değeri ile en düşük değere sahiptir. Çizelge 6.6’dan en yüksek ağırlık değerine sahip olan 6 öznitelik {24, 27, 4, 30, 29, 31} seçilmektedir. Böylece en yüksek ayırt etme kabiliyetine sahip olan öznitelikler Relief yöntemi tarafından elde edilir.

Çizelge 6.4. WDBC veri kümesi nitelikleri için Relief değerleri

Nitelik	3	4	5	6	7	8	9
Relief	0,0005	0,0116	-0,0006	-0,0005	-0,0052	-0,0099	0,0027
Nitelik	10	11	12	13	14	15	16
Relief	0,0057	-0,0124	-0,0137	0,0026	-0,0026	0,0015	0,0000
Nitelik	17	18	19	20	21	22	23
Relief	-0,0107	-0,0049	0,0009	-0,0017	0,0002	-0,0057	0,0063
Nitelik	24	25	26	27	28	29	30
Relief	0,0207	0,0055	0,0032	0,0165	-0,0029	0,0080	0,0083
Nitelik	31	32					
Relief	0,0071	-0,0011					

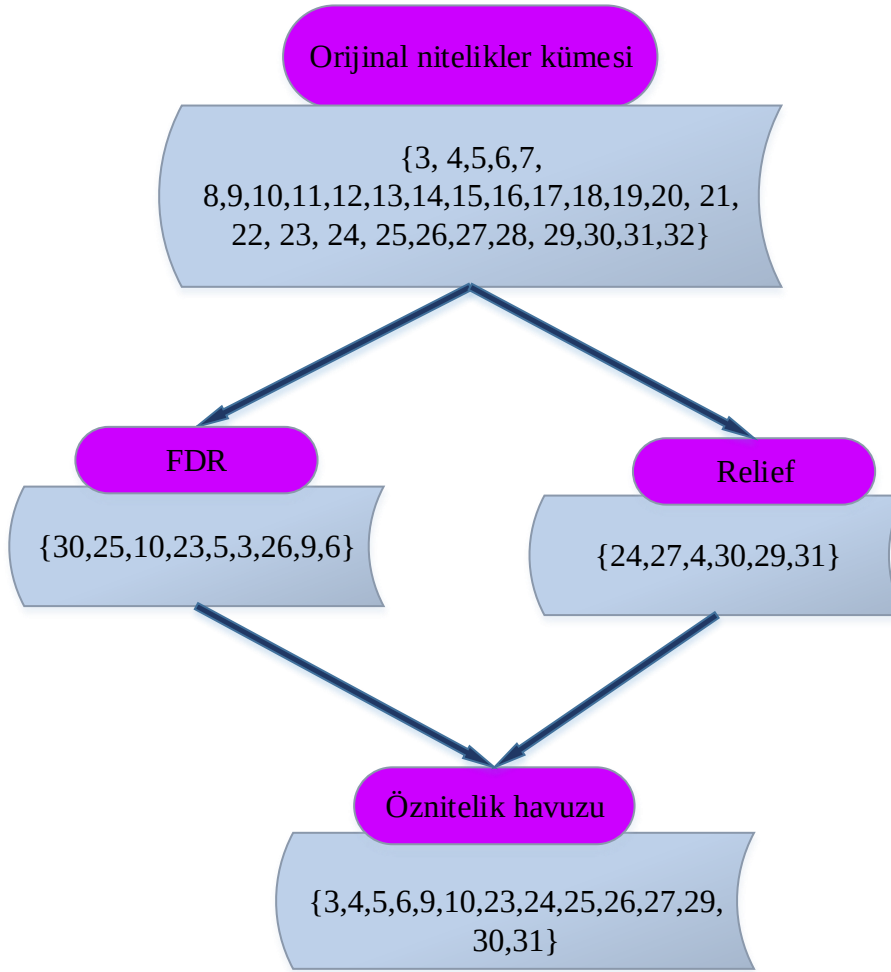
Çizelge 6.5. WDBC veri kümesinin niteliklerinin sıralanmış Relief değerleri

Nitelik	24	27	4	30	29	31	23
Relief	0,0207	0,0165	0,0116	0,0083	0,0080	0,0071	0,0063
Nitelik	10	25	26	9	13	15	19
Relief	0,0057	0,0055	0,0032	0,0027	0,0026	0,0015	0,0009
Nitelik	3	21	16	6	5	32	20
Relief	0,0005	0,0002	0,0000	-0,0005	-0,0006	-0,0011	-0,0017
Nitelik	14	28	18	7	22	8	17
Relief	-0,0026	-0,0029	-0,0049	-0,0052	-0,0057	-0,0099	-0,0107
Nitelik	11	12					
Relief	-0,0124	-0,0137					

Çizelge 6.6. WDBC veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi
FDR	9	30, 25, 10, 23, 5, 3, 26, 9, 6
Relief	6	24, 27, 4, 30, 29, 31

FDR ve Relief yöntemleri tarafından seçilen öznitelikler, öznitelik havuzunda yer alır (Şekil 6.5). Böylece orijinal verinin tüm niteliklerini kullanmak yerine, verimli ve değerli öznitelikleri seçmeye karar veririz. Bu aşama öznitelik sayısını 30 nitelikten 14 özniteliğe indirerek GA algoritmanın daha hızlı ve etkin çalışmasına yardımcı olmaktadır. Bu aşamanın sonucunda bir öznitelik havuzu {3, 4, 5, 6, 9, 10, 23, 24, 25, 26, 27, 29, 30, 31} oluşur. Sonra, GA için ilk popülasyon elde edilir. Bu popülasyon, öznitelik havuzundan rastgele oluşturulur. Bu çalışmada, popülasyon büyüklüğü 100 kabul edilmiştir. Popülasyonun her bireyinde (kromozom) seçilen öznitelik 1 değeri ve seçilmeyen 0 değeri almaktadır.



Şekil 6.5. WDBC veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması

SPECTF

Çizelge 6.7’de tüm nitelik ve FDR değerleri görülmektedir. Çizelge 6.8’de 41’inci nitelik 0,5501 FDR değeri ile en yüksek değere ve 12’nci nitelik 0,0064 değeri ile en düşük değere sahiptir. Çizelge 6.11’den görüldüğü gibi en yüksek FDR değerine sahip olan 8 öznitelik {41, 43, 27, 44, 45, 7, 42, 31} seçilmektedir.

Çizelge 6.7. SPEFCT veri kümesi nitelikleri için FDR değeri

Nitelik	2	3	4	5	6	7	8
FDR	0,0218	0,1029	0,0836	0,1995	0,1149	0,4117	0,0677
Nitelik	9	10	11	12	13	14	15
FDR	0,1449	0,0359	0,1378	0,0064	0,0551	0,0935	0,2920
Nitelik	16	17	18	19	20	21	22
FDR	0,2382	0,2563	0,0389	0,0505	0,0440	0,0732	0,0255
Nitelik	23	24	25	26	27	28	29
FDR	0,1292	0,1237	0,2583	0,3513	0,4784	0,0118	0,0748
Nitelik	30	31	32	33	34	35	36
FDR	0,2324	0,3654	0,0970	0,2662	0,1029	0,1738	0,0654
Nitelik	37	38	39	40	41	42	43
FDR	0,1403	0,0087	0,0590	0,3248	0,5501	0,3835	0,5203
Nitelik	44	45					
FDR	0,4565	0,4382					

Çizelge 6.8. SPEFCT veri kümesinin niteliklerinin sıralanmış FDR değerleri

Nitelik	41	43	27	44	45	7	42
FDR	0,5501	0,5203	0,4784	0,4565	0,4382	0,4117	0,3835
Nitelik	31	26	40	15	33	25	17
FDR	0,3654	0,3513	0,3248	0,2920	0,2662	0,2583	0,2563
Nitelik	16	30	5	35	9	37	11
FDR	0,2382	0,2324	0,1995	0,1738	0,1449	0,1403	0,1378
Nitelik	23	24	6	34	3	32	14
FDR	0,1292	0,1237	0,1149	0,1029	0,1029	0,0970	0,0935
Nitelik	4	29	21	8	36	39	13
FDR	0,0836	0,0748	0,0732	0,0677	0,0654	0,0590	0,0551
Nitelik	19	20	18	10	22	2	28
FDR	0,0505	0,0440	0,0389	0,0359	0,0255	0,0218	0,0118
Nitelik	38	12					
FDR	0,0087	0,0064					

Çizelge 6.9’de tüm nitelikler ve Relief değerlerini görülmektedir. Çizelge 6.10, nitelik ağırlık değerlerini azalan bir şekilde sıralanmış halini göstermektedir. 12’nci nitelik -0,0170 değeri ile en yüksek değere ve 42’nci nitelik-0,0398 değeri ile en düşük değere sahiptir. Çizelge 6.11’den en yüksek ağırlık değerine sahip olan 8 öz nitelik {12, 3, 24, 32, 36, 11, 39, 33} seçilmektedir.

Çizelge 6.9. SPEFCT veri kümesi nitelikleri için Relief değerleri

Nitelik	2	3	4	5	6	7	8
Relief	-0,0280	-0,0184	-0,0249	-0,0357	-0,0266	-0,0284	-0,0300
Nitelik	9	10	11	12	13	14	15
Relief	-0,0297	-0,0231	-0,0217	-0,0170	-0,0239	-0,0318	-0,0378
Nitelik	16	17	18	19	20	21	22
Relief	-0,0281	-0,0321	-0,0267	-0,0231	-0,0240	-0,0246	-0,0255
Nitelik	23	24	25	26	27	28	29
Relief	-0,0325	-0,0198	-0,0346	-0,0348	-0,0317	-0,0269	-0,0382
Nitelik	30	31	32	33	34	35	36
Relief	-0,0260	-0,0291	-0,0198	-0,0228	-0,0241	-0,0258	-0,0206
Nitelik	37	38	39	40	41	42	43
Relief	-0,0320	-0,0296	-0,0219	-0,0340	-0,0249	-0,0398	-0,0369
Nitelik	44	45					
Relief	-0,0283	-0,0317					

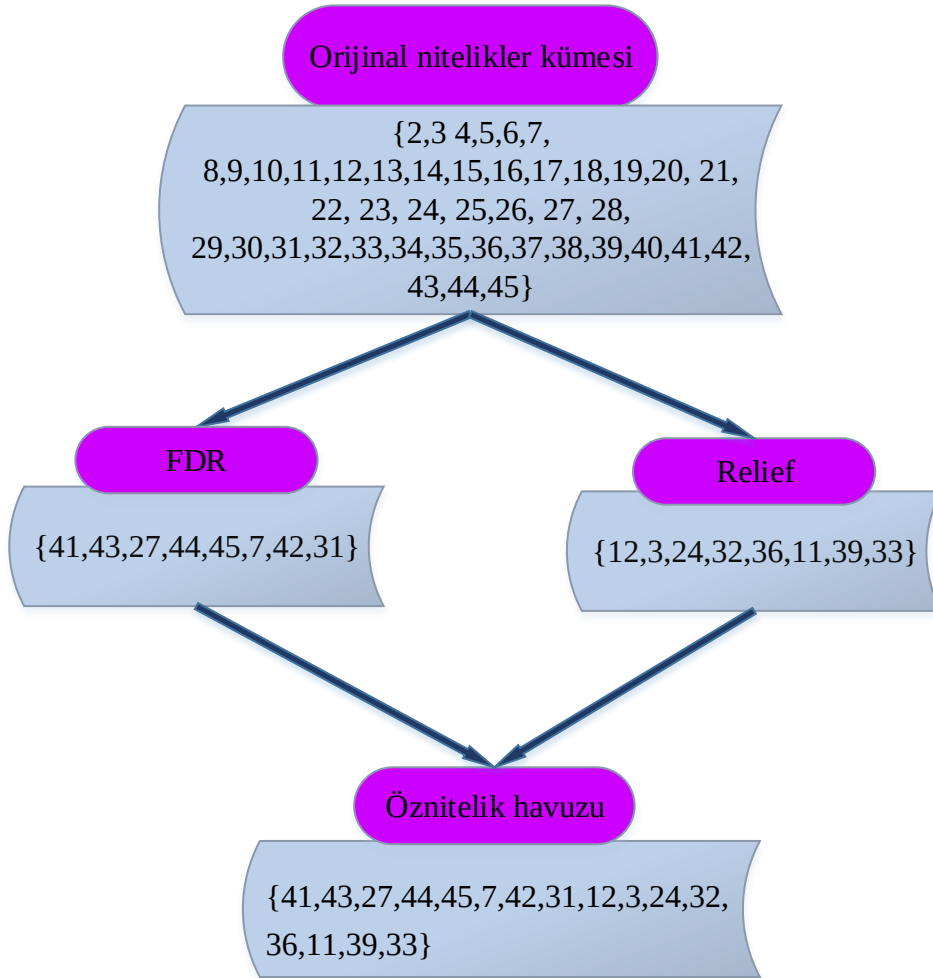
Çizelge 6.10. SPEFCT veri kümesinin niteliklerinin sıralanmış Relief değerleri

Nitelik	12	3	24	32	36	11	39
Relief	-0,0170	-0,0184	-0,0198	-0,0198	-0,0206	-0,0217	-0,0219
Nitelik	33	19	10	13	20	34	21
Relief	-0,0228	-0,0231	-0,0231	-0,0239	-0,0240	-0,0241	-0,0246
Nitelik	4	41	22	35	30	6	18
Relief	-0,0249	-0,0249	-0,0255	-0,0258	-0,0260	-0,0266	-0,0267
Nitelik	28	2	16	44	7	31	38
Relief	-0,0269	-0,0280	-0,0281	-0,0283	-0,0284	-0,0291	-0,0296
Nitelik	9	8	45	27	14	37	17
Relief	-0,0297	-0,0300	-0,0317	-0,0317	-0,0318	-0,0320	-0,0321
Nitelik	23	40	25	26	5	43	15
Relief	-0,0325	-0,0340	-0,0346	-0,0348	-0,0357	-0,0369	-0,0378
Nitelik	29	42					
Relief	-0,0382	-0,0398					

Çizelge 6.11. SPEFCT veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi
FDR	8	41, 43, 27, 44, 45, 7, 42, 31
Relief	8	12, 3, 24, 32, 36, 11, 39, 33

FDR ve Relief yöntemleri tarafından seçilen öznitelikler, öznitelik havuzunda yer alır (Şekil 6.6). Böylece orijinal verinin tüm özniteliklerini kullanmak yerine, verimli ve değerli öznitelikleri seçmeye karar veririz. Bu aşamada, öznitelik sayısını 44 nitelikten 16 öznitelığe indirerek GA algoritmanın daha hızlı ve etkin çalışmasına yardımcı olmaktadır. Bu aşamanın sonucunda bir öznitelik havuzu {41, 43, 27, 44, 45, 7, 42, 31, 12, 3, 24, 32, 36, 11, 39, 33} oluşur.



Şekil 6.6. SPEFCT veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması

Statlog kalp

Çizelge 6.12’de tüm nitelik ve FDR değerleri görülmektedir. Çizelge 6.13’de, 13’üncü nitelik 0,7540 FDR değeri ile en yüksek değere ve 6’ıncı nitelik 0,0005 değeri ile en düşük değere sahiptir. Çizelge 6.16’dan, gösterildiği gibi en yüksek FDR değerine sahip olan 7 öznitelik {13, 12, 3, 8, 9, 10, 11} seçilmektedir.

Çizelge 6.12. Statlog veri kümesi nitelikleri için FDR değeri

Nitelik	1	2	3	4	5	6	7
FDR	0,0966	0,2016	0,4323	0,0489	0,0288	0,0005	0,0691
Nitelik	8	9	10	11	12	13	
FDR	0,4182	0,4138	0,4046	0,2599	0,5001	0,7540	

Çizelge 6.13. Statlog veri kümesinin niteliklerinin sıralanmış FDR değerleri

Nitelik	13	12	3	8	9	10	11
FDR	0,7540	0,5001	0,4323	0,4182	0,4138	0,4046	0,2599
Nitelik	2	1	7	4	5	6	
FDR	0,2016	0,0966	0,0691	0,0489	0,0288	0,0005	

Çizelge 6.14’de, tüm nitelik ve Relief değerleri görülmektedir. Çizelge 6.15, nitelik ağırlık değerlerini azalan bir şekilde sıralanmış halini göstermektedir. 12’nci nitelik -0,0460 değeri ile en yüksek değere ve 2’nci nitelik -0,0108 değeri ile en düşük değere sahiptir. Çizelge 6.16’dan, en yüksek ağırlık değerine sahip olan 7 öz nitelik {12, 3, 11, 13, 10, 8, 4} seçilmektedir.

Çizelge 6.14. Statlog veri kümesi nitelikleri için Relief değerleri

Nitelik	1	2	3	4	5	6	7
Relief	0,0060	-0,0108	0,0418	0,0107	-0,0065	-0,0051	0,0096
Nitelik	8	9	10	11	12	13	
Relief	0,0125	-0,0057	0,0153	0,0216	0,0460	0,0199	

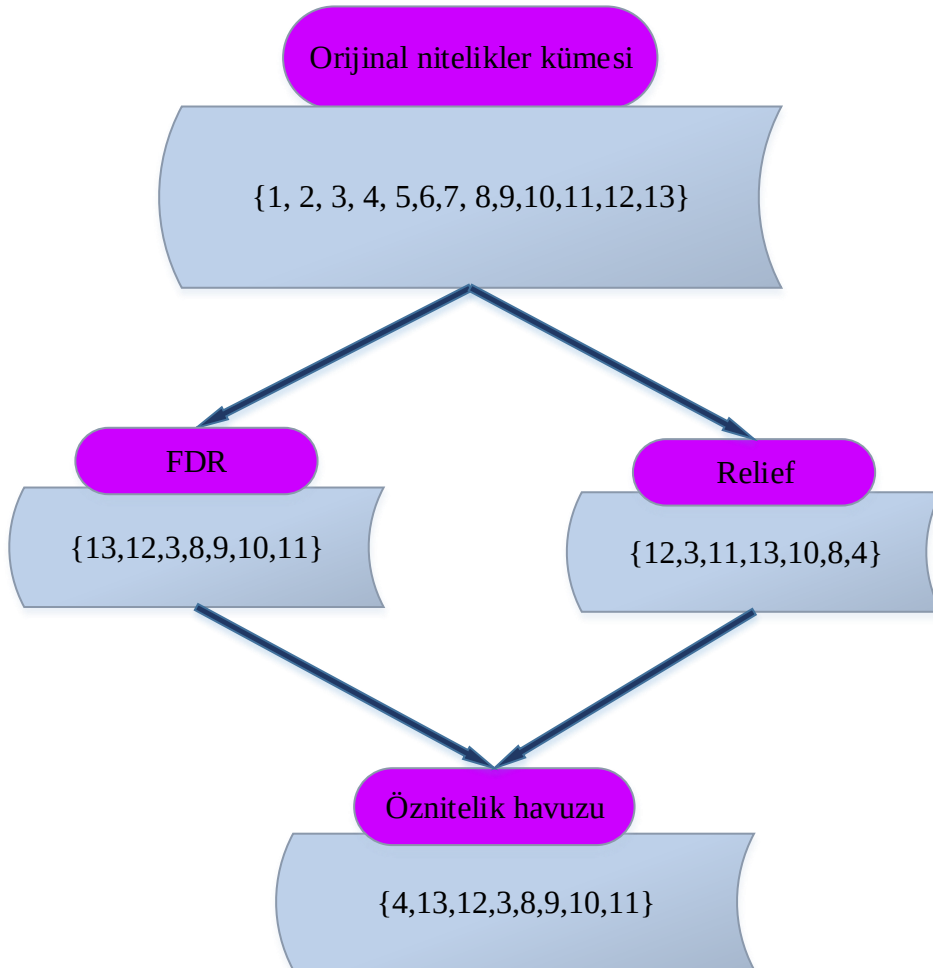
Çizelge 6.15. Statlog veri kümesinin niteliklerinin sıralanmış Relief değerleri

Nitelik	12	3	11	13	10	8	4
Relief	0,0460	0,0418	0,0216	0,0199	0,0153	0,0125	0,0107
Nitelik	7	1	6	9	5	2	
Relief	0,0096	0,0060	-0,0051	-0,0057	-0,0065	-0,0108	

Çizelge 6.16. Statlog veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi
FDR	7	13, 12, 3, 8, 9, 10, 11
Relief	7	12, 3, 11, 13, 10, 8, 4

FDR ve Relief yöntemleri tarafından seçilen öznitelikler, öznitelik havuzunda yer alır(Şekil 6.7). Böylece orijinal verinin tüm niteliklerini kullanmak yerine, verimli ve değerli öznitelikleri seçmeye karar veririz. Bu aşamada, öznitelik sayısını 13 nitelikten 8 özniteliğe indirerek GA algoritmanın daha hızlı ve etkin çalışmasına yardımcı olmaktadır. Bu aşamanın sonucunda bir öznitelik havuzu {13, 12, 3, 8, 9, 10, 11, 4} oluşur.



Şekil 6.7. Statlog veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması

WPBC

Çizelge 6.17, tüm nitelik ve FDR değerlerini göstermektedir. Çizelge 6.18’de, 3’üncü nitelik 0,4522 FDR değeri ile en yüksek değere ve 9’üncü nitelik 0,0004 değeri ile en düşük değere sahiptir. Çizelge 6.21’den gösterdiği gibi en yüksek FDR değerine sahip olan 5 öz nitelik {3, 24, 26, 27, 7} seçilmektedir.

Çizelge 6.17. WPBC veri kümesi nitelikleri için FDR değeri

Nitelik	3	4	5	6	7	8	9
FDR	0,4522	0,0715	0,0148	0,0720	0,0817	0,0005	0,0004
Nitelik	10	11	12	13	14	15	16
FDR	0,0051	0,0295	0,0334	0,0438	0,0373	0,0190	0,0407
Nitelik	17	18	19	20	21	22	23
FDR	0,0473	0,0122	0,0008	0,0162	0,0200	0,0047	0,0069
Nitelik	24	25	26	27	28	29	30
FDR	0,1280	0,0076	0,1218	0,1217	0,0038	0,0018	0,0005
Nitelik	31	32	33	34	35		
FDR	0,0142	0,0119	0,0092	0,0814	0,0745		

Çizelge 6.18. WPBC veri kümesinin niteliklerinin sıralanmış FDR değerleri

Nitelik	3	24	26	27	7	34	35
FDR	0,4522	0,1280	0,1218	0,1217	0,0817	0,0814	0,0745
Nitelik	6	4	17	13	16	14	12
FDR	0,0720	0,0715	0,0473	0,0438	0,0407	0,0373	0,0334
Nitelik	11	21	15	20	5	31	18
FDR	0,0295	0,0200	0,0190	0,0162	0,0148	0,0142	0,0122
Nitelik	32	33	25	23	10	22	28
FDR	0,0119	0,0092	0,0076	0,0069	0,0051	0,0047	0,0038
Nitelik	29	19	8	30	9		
FDR	0,0018	0,0008	0,0005	0,0005	0,0004		

Çizelge 6.19, tüm nitelikler ve Relief değerlerini göstermektedir. Çizelge 6.20, nitelik ağırlık değerlerini azalan bir şekilde sıralanmış halini göstermektedir ki 35'inci nitelik 0,0162 değeri ile en yüksek değere ve 33'üncü nitelik -0,0120 değer ile en düşük değere sahiptir. Çizelge 6.21'dan, en yüksek ağırlık değerine sahip olan 12 öz nitelik {35, 3, 25, 34, 14, 16, 17, 5, 11, 23, 12, 13} seçilmektedir.

Çizelge 6.19. WPBC veri kümesi nitelikleri için Relief değerleri

Nitelik	3	4	5	6	7	8	9
Relief	0.0092	-0.0057	0.0042	-0.0065	-0.0042	-0.0055	-0.0030
Nitelik	10	11	12	13	14	15	16
Relief	-0.0029	0.0036	0.0013	0.0002	0.0082	-0.0036	0.0070
Nitelik	17	18	19	20	21	22	23
Relief	0.0069	-0.0072	0.0000	-0.0018	-0.0021	-0.0040	0.0020
Nitelik	24	25	26	27	28	29	30
Relief	-0.0053	0.0088	-0.0061	0.0002	-0.0051	-0.0101	-0.0028
Nitelik	31	32	33	34	35		
Relief	-0.0080	-0.0076	-0.0120	0.0086	0.0162		

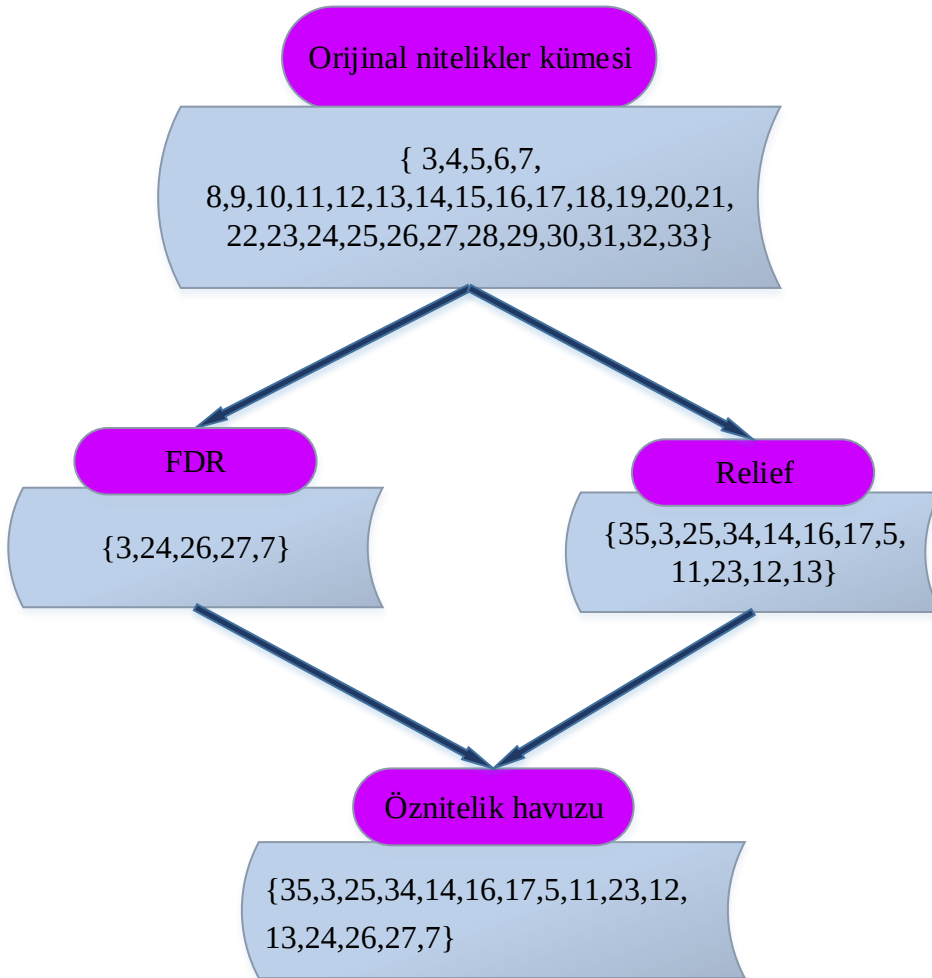
Çizelge 6.20. WPBC veri kümesinin niteliklerinin sıralanmış Relief değerleri

Nitelik	35	3	25	34	14	16	17
Relief	0.0162	0.0092	0.0088	0.0086	0.0082	0.0070	0.0069
Nitelik	5	11	23	12	13	27	19
Relief	0.0042	0.0036	0.0020	0.0013	0.0002	0.0002	0.0000
Nitelik	20	21	30	10	9	15	22
Relief	-0.0018	-0.0021	-0.0028	-0.0029	-0.0030	-0.0036	-0.0040
Nitelik	7	28	24	8	4	26	6
Relief	-0.0042	-0.0051	-0.0053	-0.0055	-0.0057	-0.0061	-0.0065
Nitelik	18	32	31	29	33		
Relief	-0.0072	-0.0076	-0.0080	-0.0101	-0.0120		

Çizelge 6.21. WPBC veri kümesinde FDR ve Relief tarafından seçilen öznitelik alt kümeleri

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi
FDR	5	3, 24, 26, 27, 7
Relief	12	35, 3, 25, 34, 14, 16, 17, 5, 11, 23, 12, 13

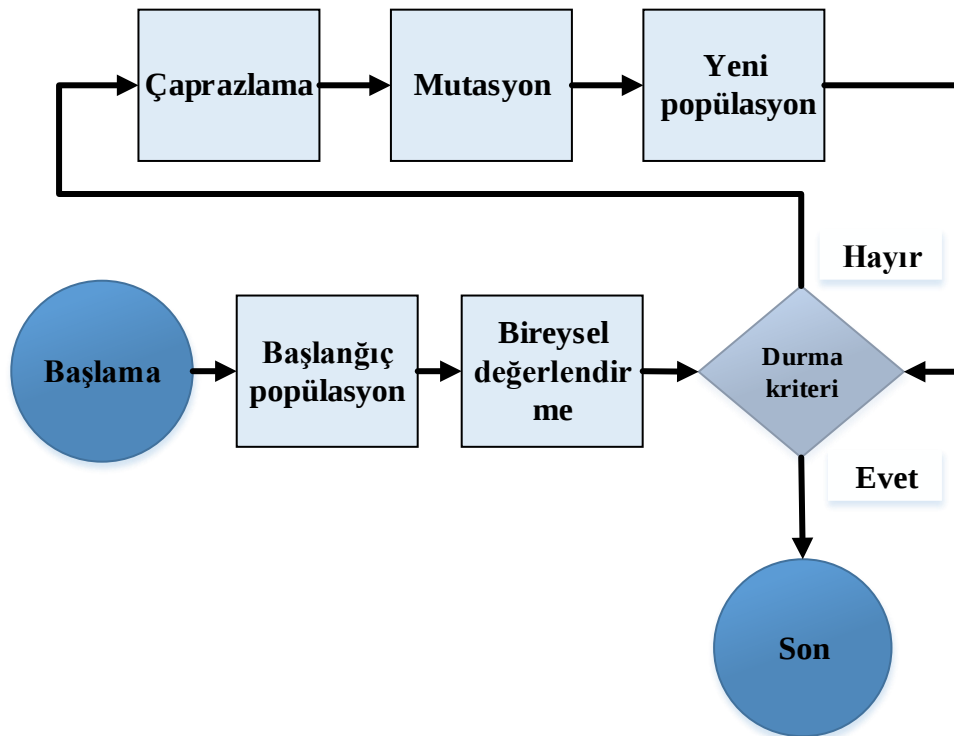
FDR ve Relief yöntemleri tarafından seçilen öznitelikler, öznitelik havuzunda yer alır (Şekil 6.8). Böylece orijinal verinin tüm niteliklerini kullanmak yerine, verimli ve değerli öznitelikleri seçmeye karar veririz. Bu aşamada, öznitelik sayısını 33 nitelikten 16 özniteliğe indirerek GA algoritmanın daha hızlı ve etkin çalışmasına yardımcı olmaktadır. Bu aşamanın sonucunda bir öznitelik havuzu {35, 3, 25, 34, 14, 16, 17, 5, 11, 23, 12, 13, 24, 26, 27, 7} oluşur.



Şekil 6.8. WPBC veri kümesinin FDR ve Relief ile öznitelik havuzunun oluşturulması

6.2.2. İkinci aşama: Genetik algoritma ile özneliklerin belirlenmesi

GA aşamasında, başlangıç popülasyonu üzerinde farklı sınıflandırma yöntemleri (DVM, DAA, NB ve KA) yürütülür. Aşağıdaki işlemler her sınıflandırma yöntemi için yapılır. Şekil 6.9, GA aşaması için genel bir bakış sağlamaktadır. İlk olarak, GA, öznelik seçimi süreci tarafından üretilen başlangıç popülasyonunu girdi olarak alır. Sonra, popülasyonun her bireyi (kromozom) GA'nın uygunluk fonksiyonu tarafından değerlendirilir. Burada durma kriteri (iterasyon sayısı) kontrol edilir. GA sonlanana kadar seçilen bireyler üzerinde çaprazlama ve mutasyon yapılır. Bu operatörler yeni bir popülasyon oluşturur ve tekrardan değerlendirme aşamasına dönüp, durma kriterine ulaşıncaya kadar bu işlemler devam eder. Son olarak, durma kriterini sağlandığında, GA, en iyi sınıflandırma doğruluğuna ve en uygun veya en uyguna yakın bir öznelik alt kümesi elde eder.



Şekil 6.9. GA modelinin blok diyagramı

Uygunluk fonksiyonu:

En yüksek başarıya sahip özneliklerin bulunması uygunluk fonksiyonunun başarısına bağlıdır. Uygunluk fonksiyonu Eşitlik 6.1’de tanımlanmıştır.

$$F(i) = ca(i) - s(i)/w \quad (6.1)$$

Eşitlik 6.1, i seçilen bir öznitelik alt kümesi (bireysel) ve w bir ağırlık parametresidir. $ca(i)$ belirlenen sınıflandırma algoritmasının sınıflandırma başarısıdır. $s(i)$, i öznitelik alt kümesinin boyutudur ve $F(i)$, i öznitelik alt kümesinin uygunluk değeridir. w belirlenmiş ağırlık değeridir.

Uygunluk fonksiyonunda, i bireyin $s(i)$ öznitelik sayısı, $ca(i)$ sınıflandırma doğruluğundan çıkarılır. Belirli bir w değeri için; i bireyin uygunluk değeri, sınıflandırma doğruluğu artarken ve bireyin boyutu azalırken artar. Öte yandan, $s(i)$ boyutunun arttırılması veya sınıflandırma doğruluğun azalması ile uygunluk değeri azalır. w değerini azaltarak bireyin boyutuna daha fazla öncelik verebiliriz ve aksi bir şekilde w değerini artırarak göz ardı edilebiliriz. Deneylerde, w için 100'ün en uygun değer olduğuna karar verilmiştir.

Seçim:

Her yineleme adımında, en yüksek uygunluk değerine sahip olan 10 birey seçilir, doğrudan yeni popülasyona atılır; en düşük değere sahip olan 10 birey çıkartılır. İlk 10 birey ve kalan bireyler için Rulet tekerleği hesaplanır. Rulet tekerleği her bireyin uygunluk değerine bağlı, muhtemel farklı boyutlardadır. Rulet tekerleği seçiminde, birey i 'in seçilme olasılığı, $P(\text{seçim} = i)$ aşağıdaki gibi hesaplanır:

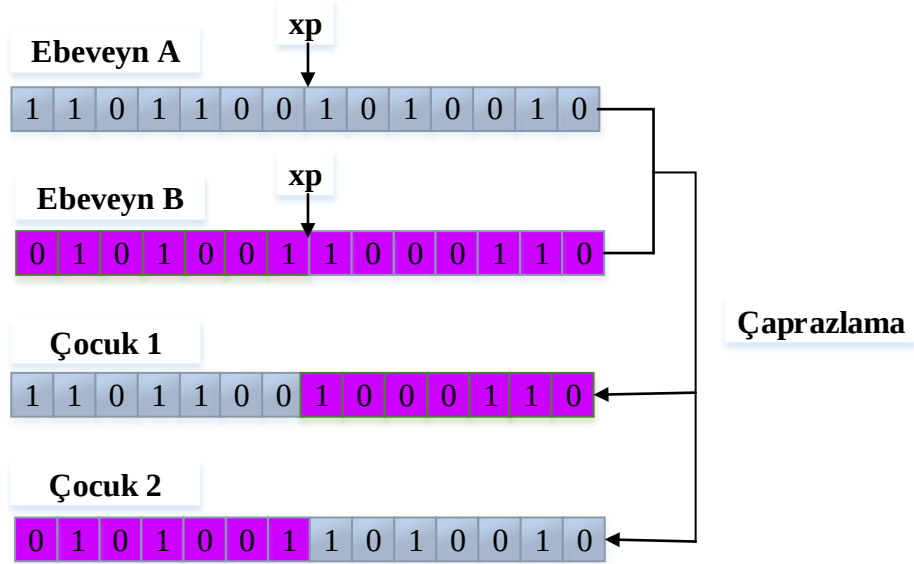
$$P(\text{seçim} = i) = \frac{Uygunluk(i)}{\sum_{j=1}^n Uygunluk(j)} \quad (6.2)$$

Her bireyin uygunluk değeri oranı, uygunluk değerlerinin toplamına oranlanır. Böylece, yüksek uygunluk değerine sahip olan bireyler, yeni popülasyona daha fazla katkıda bulunmaktadır ve daha düşük uygunluk değerine sahip olan bireyler daha az etkiye sahip olmaktadır.

Çaprazlama:

Burada tek nokta çaprazlama operatörünü kullanmaktayız. Çaprazlama noktasının x_p olduğu varsayılarak, bu değer rastgele seçilmiştir. Bu operatör iki yeni birey oluşturur. İlk

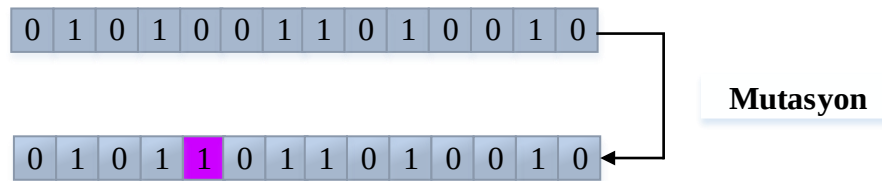
çocukta, ilk xp bit ilk ebeveynden ve kalan bitler ikinci ebeveynden oluşmaktadır. İkinci çocukta, ilk xp bit, ikinci ebeveynden ve kalan bitler ilk ebeveyn tarafından oluşmaktadır. Çaprazlama işlemi Şekil 6.10'de gösterilmiştir.



Şekil 6.10. Çaprazlama işlevi

Mutasyon:

Mutasyona uğratılmak istenen bireyler rastgele seçilir ve rastgele seçilmiş bireylerin bir biti rastgele değiştirilir. Mutasyon işlemi Şekil 6.11'de gösterilmiştir.



Şekil 6.11. Mutasyon işlemi

6.3. Sınıflandırma

6.3.1. WDBC veri kümesi

DVM

Bu aşamada, her tekrar için her bireyin sınıflandırma doğruluğu hesaplanır, Çizelge 6.22’de her iterasyonda sınıflandırma doğruluğu en yüksek olan bireyin doğruluğunu göstermektedir. Çizelge 6.22’de görebileceğimiz gibi, DVM ilk tekrarda 91,41% doğruluk ile başlar, sonra ikinci iterasyonda 92,98% çoğalır, üçüncü ve dördüncü iterasyonlarda sırasıyla 94,36% ve 99,41% elde eder, beşinci iterasyonda 100% doğruluğa ulaşır ve bu doğruluk ile 25’inci iterasyona kadar devam eder. Öznitelik sayısı beşinci iterasyonda, sekizinci iterasyondan fazla olduğu için en iyi alt küme olarak seçilmez. Sekizinci iterasyonda, DVM, 5 öznitelikten oluşan {4, 9, 26, 27, 31} alt kümesi ile 100% sınıflandırma doğruluğu elde eder. GA için uygunluk fonksiyonu DVM kullanıldığında aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.22.WDBC veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	91,41	92,98	94,36	99,41	100	100	100
İterasyon	8	9	10	11	12	13	14
Doğruluk	100	100	100	100	100	100	100
İterasyon	15	16	17	18	19	20	21
Doğruluk	100	100	100	100	100	100	100
İterasyon	22	23	24	25			
Doğruluk	100	100	100	100			

Aşağıda hata matrisi ve ondan hesaplanan performans ölçütleri gösterilmiştir (Şekil 6.12).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	63 (DP)	0 (YP)
	N	0 (YN)	107 (DN)

Şekil 6.12. WDBC veri kümesinde DVM sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{63 + 107}{63 + 0 + 107 + 0} = 1$$

$$\text{Hata oranı} = \frac{0 + 0}{63 + 0 + 107 + 0} = 0$$

$$\text{Doğru pozitif oranı} = \frac{63}{63 + 0} = 1$$

$$\text{Yanlış pozitif oranı} = \frac{0}{0 + 107} = 0$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 1$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0 = 1$$

$$\text{hassasiyet} = \frac{63}{63 + 0} = 1$$

$$\text{yeniden çağırma} = \frac{63}{63} = 1$$

$$F - \text{ölçü} = \frac{2}{1/1 + 1/1} = 1$$

DAA

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, DAA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.23'de DAA ilk iterasyonda 89,85% doğruluk ile başlar ve ikinci iterasyonda 91,33% yükselir, 3'üncü, 4'üncü, 5'nci ve 6'ncı iterasyonlarda sırayla 93,00%, 95,17%, 98,56% ve 98,93% doğrulukları elde etmektedir. 7'nci ve 8'ninci iterasyonlarda büyük bir artışla 99,15% ve 99,41% doğruluğunu sağlar. Daha sonra 9'üncü iterasyonda 6 öznelikten oluşan {10, 24, 25, 26, 27, 31} alt kümesi ile 99,41% sınıflandırma doğruluğunu elde eder.

Çizelge 6.23. WDBC veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	89,85	91,33	93,00	95,17	98,56	98,93	99,15
İterasyon	8	9	10	11	12	13	14
Doğruluk	99,41	99,41	99,41	99,41	99,41	99,41	99,41
İterasyon	15	16	17	18	19	20	21
Doğruluk	99,41	99,41	99,41	99,41	99,41	99,41	99,41
İterasyon	22	23	24	25			
Doğruluk	99,41	99,41	99,41	99,41			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.13).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	62 (DP)	1 (YP)
	N	0 (YN)	107 (DN)

Şekil 6.13. WDBC veri kümesinde DAA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{62 + 107}{62 + 1 + 107 + 0} = 0,994$$

$$\text{Hata oranı} = \frac{1 + 0}{62 + 1 + 107 + 0} = 0,0058$$

$$\text{Doğru pozitif oranı} = \frac{62}{62 + 0} = 1$$

$$\text{Yanlış pozitif oranı} = \frac{1}{1 + 107} = 0,0092$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 1$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,0092 = 0,9908$$

$$\text{hassasiyet} = \frac{62}{62 + 0} = 1$$

$$\text{yeniden çağırma} = \frac{62}{62} = 1$$

$$\text{Doğruluk} = \frac{62 + 107}{63 + 107} = 0,994$$

$$F - \text{ölçü} = \frac{2}{1/1 + 1/1} = 1$$

NB

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, NB'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.24'de görüldüğü gibi, NB yöntemi ilk tekrarda 90,12% sınıflandırma doğruluk ile başlar ve yükselmeye devam eder böyle ki ikinci iterasyonda 91,13% doğruluğa yükselir, üçüncü, dördüncü ve beşinci iterasyonlarda sırayla 92,58%, 94,69% ve 96,14%

doğrulukları elde eder. Altıncı ve yedinci iterasyonlarda büyük bir artışla, 97,27% ve 98,50% doğrulukları sağlar. Sekizinci iterasyonlarda, 98,82% doğruluğa değişip ve sonra 21'inci iterasyona kadar devam eder, sonra 22'nci iterasyonda 99,41% doğruluğuna artar ve bu değerle devam eder. NB, 22'nci iterasyonda 8 öznitelik {3, 4, 10, 23, 24, 26, 27, 31} ile 99,4118% doğruluğa ulaşır.

Çizelge 6.24. WDBC veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	90,12	91,13	92,58	94,69	96,14	97,27	98,50
İterasyon	8	9	10	11	12	13	14
Doğruluk	98,82	98,82	98,82	98,82	98,82	98,82	98,82
İterasyon	15	16	17	18	19	20	21
Doğruluk	98,82	98,82	98,82	98,82	98,82	98,82	98,82
İterasyon	22	23	24	25			
Doğruluk	99,41	99,41	99,41	99,41			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.14).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	62 (DP)	1 (YP)
	N	0 (YN)	107 (DN)

Şekil 6.14. WDBC veri kümesinde NB sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{62 + 107}{62 + 1 + 107 + 0} = 0,994$$

$$\text{Hata oranı} = \frac{1 + 0}{62 + 1 + 107 + 0} = 0,0058$$

$$\text{Doğru pozitif oranı} = \frac{62}{62 + 0} = 1$$

$$\text{Yanlış pozitif oranı} = \frac{1}{1 + 107} = 0,0092$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 1$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,0092 = 0,9908$$

$$\text{hassasiyet} = \frac{62}{62 + 0} = 1$$

$$\text{yeniden çağırma} = \frac{62}{62} = 1$$

$$F - \text{ölçü} = \frac{2}{1/1 + 1/1} = 1$$

Karar Ağacı

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, KA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.25'de KA, ilk iterasyonda 88,93% doğruluğunu gösterir, 8'inci iterasyona kadar hızlı ve yükselen bir şekilde artıp ve sonra 9'uncü iterasyonda 97,00% doğruluğa ulaşır. KA, 10'uncü iterasyonda 5 öznitelik {3, 4, 6, 10, 30} ile 97,06% sınıflandırma doğruluğuna ulaşır ve 25'inci itrasyona kadar bu doğrulukla devam eder.

Çizelge 6.25. WDBC veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	88,93	90,98	91,56	93,00	94,78	95,69	96,10
İterasyon	8	9	10	11	12	13	14
Doğruluk	96,90	97,00	97,06	97,06	97,06	97,06	97,06
İterasyon	15	16	17	18	19	20	21
Doğruluk	97,06	97,06	97,06	97,06	97,06	97,06	97,06
İterasyon	22	23	24	25			
Doğruluk	97,06	97,06	97,06	97,06			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.15).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	60 (DP)	3 (YP)
	N	2 (YN)	105 (DN)

Şekil 6.15. WDBC veri kümesinde KA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{60 + 105}{60 + 3 + 105 + 2} = 0,9705$$

$$\text{Hata oranı} = \frac{3 + 2}{60 + 3 + 105 + 2} = 0,0294$$

$$\text{Doğru pozitif oranı} = \frac{60}{60 + 2} = 0,9677$$

$$\text{Yanlış pozitif oranı} = \frac{3}{3 + 105} = 0,0277$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,9677$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,0227 = 0,9773$$

$$\text{hassasiyet} = \frac{60}{60 + 2} = 0,9677$$

$$\text{yeniden çağırma} = \frac{60}{62} = 0,9677$$

$$F - \text{ölçü} = \frac{2}{1/0,9677 + 1/0,9677} = 0,9677$$

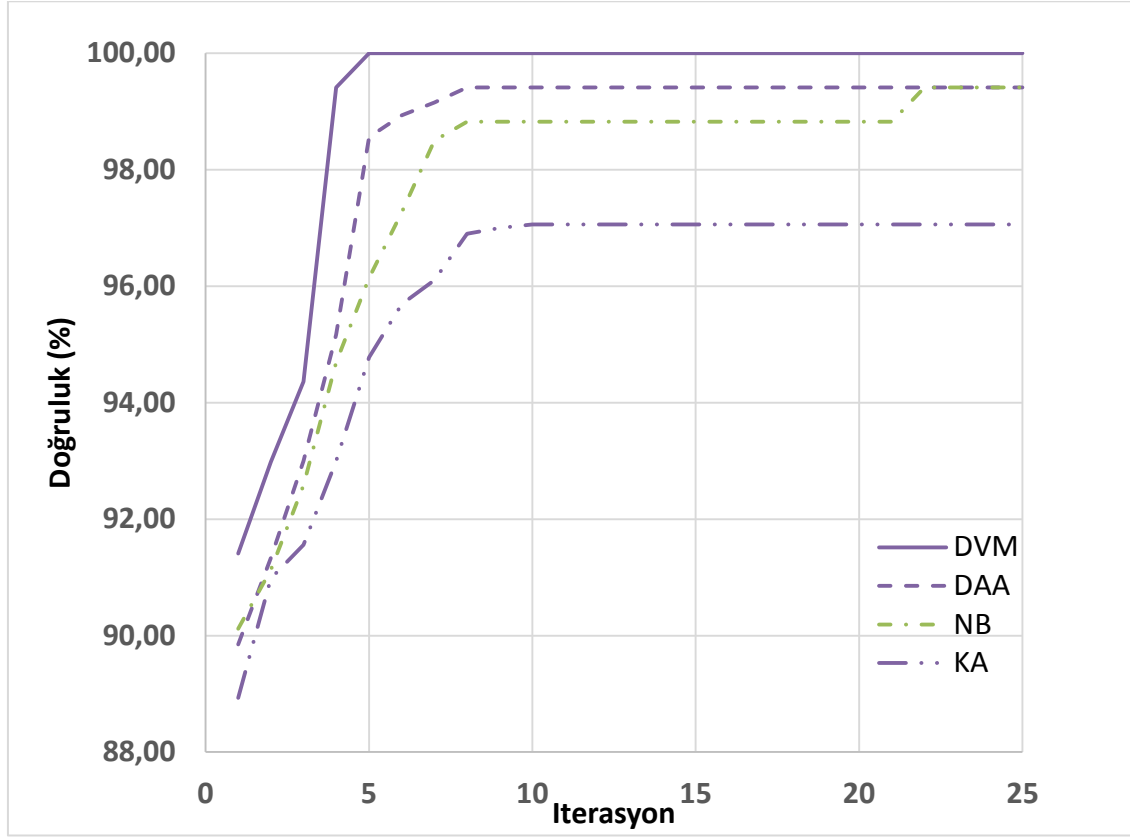
Çizelge 6.26'de DVM, DAA, NB ve KA yöntemlerin öznitelik alt kümeleri ve doğrulukları bir arada gösterilmektedir. Çizelge 6.26'de gösterilen yöntemleri karşılaştırıldığında, DVM az sayıda öznitelik ile daha yüksek sınıflandırma performansı elde etmektedir. DVM, 5 öznitelik {4, 9, 26, 27, 31} ve 100% doğruluk ile en başarılı yöntemdir. DAA yöntemi 6 öznitelik {10, 24, 25, 26, 27, 31} ve 99,4118% başarı ile ikinci sırada yer almaktadır. Üçüncü sırada NB, 8 öznitelik {3, 4, 10, 23, 24, 26, 27, 31} ve 99,4118% doğruluk ile yer alır. En sonda KA, 5 öznitelik {3, 4, 6, 10, 30} ve 97,0588% doğrulukla yer almaktadır. DAA ve NB 99, 4118% doğruluk ile aynı sonuçlar elde eder ama DAA, NB ye göre daha az öznitelik sayısı ile sınıflandırma işlevini gerçekleştirir.

Çizelge 6.26. WDBC veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi	Doğruluk (%)
DVM	5	4, 9, 26, 27, 31	100
DAA	6	10, 24, 25, 26, 27, 31	99,4118
NB	8	3, 4, 10, 23, 24, 26, 27, 31	99,4118
KA	5	3, 4, 6, 10, 30	97,0588

Önerilen hibrit genetik algoritma yönteminde 4 sınıflandırıcı yöntemi 25 iterasyon için denenmiştir ve bu denemede her yöntemin sınıflandırma doğruluğu her iterasyon için detaylı bir şekilde incelenmiştir. Şekil 6.16, 25 iterasyona dayalı DVM, DAA, NB ve KA

sınıflandırma doğruluklarının eğimlerini göstermektedir.



Şekil 6.16. WDBC veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırma doğrulukları

6.3.2. SPECTF veri kümesi

DVM

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, DVM'nin doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Her tekrarlama her bireyin sınıflandırma doğruluğu hesaplanır, Çizelge 6.27'de her iterasyonda sınıflandırma doğruluğu en yüksek olan bireyin doğruluğunu göstermektedir. Çizelge 6.27'de de görebileceğimiz gibi, DVM ilk tekrarda 82,5% doğruluk ile başlar, sonra ikinci iterasyonda 83,75% çoğalır, üçüncü iterasyonda 85% elde eder, dördüncü iterasyonda 87,5% sonuçlar, beşinci iterasyonda 88,75% doğruluğa ulaşır ve bu doğruluk ile 25'inci iterasyona kadar devam eder. Beşinci iterasyonda, DVM, 10 öznitelikten oluşan

{3, 11, 12, 24, 32, 33, 39, 41, 44, 45} alt kümesi ile 88,75% sınıflandırma doğruluğu elde eder.

Çizelge 6.27. SPECTF veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	82,5	83,75	85	87,5	88,75	88,75	88,75
İterasyon	8	9	10	11	12	13	14
Doğruluk	88,75	88,75	88,75	88,75	88,75	88,75	88,75
İterasyon	15	16	17	18	19	20	21
Doğruluk	88,75	88,75	88,75	88,75	88,75	88,75	88,75
İterasyon	22	23	24	25			
Doğruluk	88,75	88,75	88,75	88,75			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.17).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	13 (DP)	2 (YP)
	N	7 (YN)	58 (DN)

Şekil 6.17. SPCTF veri kümesinde DVM sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{13 + 58}{13 + 2 + 58 + 7} = 0,8875$$

$$\text{Hata oranı} = \frac{2 + 7}{13 + 2 + 58 + 7} = 0,1125$$

$$\text{Doğru pozitif oranı} = \frac{13}{13 + 7} = 0,65$$

$$\text{Yanlış pozitif oranı} = \frac{2}{2 + 58} = 0,0333$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,65$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,033 = 0,9667$$

$$\text{hassasiyet} = \frac{13}{13 + 7} = 0,65$$

$$\text{yeniden çağırma} = \frac{13}{20} = 0,65$$

$$\text{Doğruluk} = \frac{13 + 58}{15 + 65} = 0,8875$$

$$F - \text{ölçü} = \frac{2}{1/0,65 + 1/0,65} = 0,65$$

DAA

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, DAA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.28'de DAA ilk iterasyonda 83,75% doğruluk ile başlar ve ikinci iterasyonda 86,25% yükselir, 3'üncü, 4'üncü iterasyonlarda sırayla 87,5%, 88,75% doğrulukları elde eder. 5'nci iterasyonda 9 öznitelikten oluşan {3, 7, 27, 31, 32, 33, 36, 44, 45} alt kümesi ile 91,25% sınıflandırma doğruluğuna ulaşır.

Çizelge 6.28. SPECTF veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	83,7500	86,25	87,5	88,75	91,25	91,25	91,25
İterasyon	8	9	10	11	12	13	14
Doğruluk	91,25	91,25	91,25	91,25	91,25	91,25	91,25
İterasyon	15	16	17	18	19	20	21
Doğruluk	91,25	91,25	91,25	91,25	91,25	91,25	91,25
İterasyon	22	23	24	25			
Doğruluk	91,25	91,25	91,25	91,25			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.18).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	13 (DP)	2 (YP)
	N	5 (YN)	60 (DN)

Şekil 6.18. SPCTF veri kümesinde DAA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{13 + 60}{13 + 2 + 60 + 5} = 0,9125$$

$$\text{Hata oranı} = \frac{2 + 5}{13 + 2 + 60 + 5} = 0,0875$$

$$\text{Doğru pozitif oranı} = \frac{13}{13 + 5} = 0,7222$$

$$\text{Yanlış pozitif oranı} = \frac{2}{2 + 60} = 0,0322$$

Duyarlılık (Sensitivity) = $DP \text{ oranı} = 0,7222$

Seçicilik (Specifity) = $1 - YP \text{ oranı} = 1 - 0,0322 = 0,9678$

$$\text{hassasiyet} = \frac{13}{13 + 5} = 0,722$$

$$\text{yeniden çağırma} = \frac{13}{18} = 0,722$$

$$\text{Doğruluk} = \frac{13 + 60}{15 + 65} = 0,9125$$

$$F - \text{ölçü} = \frac{2}{1/0,722 + 1/0,722} = 0,722$$

NB

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, NB'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.29'de görüldüğü gibi, NB yöntemi ilk tekrarda 78,75% sınıflandırma doğruluk ile başlar ve yükselmeye devam eder böyle ki ikinci iterasyonda 80% doğruluğuna yükselir, üçüncü iterasyonda 81,25% doğruluğu elde eder. NB, üncü iterasyonda bir öznitelik {36} ile 81,25% doğruluğa ulaşır.

Çizelge 6.29. SPECTF veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	78,75	80	81,2500	81,2500	81,2500	81,2500	81,2500
İterasyon	8	9	10	11	12	13	14
Doğruluk	81,2500	81,2500	81,2500	81,2500	81,2500	81,2500	81,2500
İterasyon	15	16	17	18	19	20	21
Doğruluk	81,2500	81,2500	81,2500	81,2500	81,2500	81,2500	81,2500
İterasyon	22	23	24	25			
Doğruluk	81,2500	81,2500	81,2500	81,2500			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.19).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	0 (DP)	15 (YP)
	N	0 (YN)	65 (DN)

Şekil 6.19. SPCTF veri kümesinde NB sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{0 + 65}{0 + 15 + 65 + 0} = 0,8125$$

$$\text{Hata oranı} = \frac{15 + 0}{0 + 15 + 65 + 0} = 0,1875$$

$$\text{Doğru pozitif oranı} = \frac{0}{0 + 0} = 0$$

$$\text{Yanlış pozitif oranı} = \frac{15}{15 + 65} = 0,1875$$

Duyarlılık (Sensitivity) = $DP \text{ oranı} = 0$

Seçicilik (Specifity) = $1 - YP \text{ oranı} = 1 - 0,1875 = 0,8125$

hassasiyet = 0

yeniden çağırma = 0

$$\text{Doğruluk} = \frac{0 + 65}{15 + 65} = 0,8125$$

F – ölçü = 0

Karar Ağacı

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, KA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.30'de KA, ilk iterasyonda 83,75% doğruluğunu gösterir, ikinci iterasyonda 85%, üçüncü iterasyonda 90% doğruluğa ulaşır. KA, üçüncü iterasyonda 5 öznitelik {31, 32, 33, 42, 43} ile 90% sınıflandırma doğruluğuna ulaşır ve 25'inci itrasyona kadar bu doğrulukla devam eder.

Çizelge 6.30. SPECTF veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	83,75	85	90	90	90	90	90
İterasyon	8	9	10	11	12	13	14
Doğruluk	90	90	90	90	90	90	90
İterasyon	15	16	17	18	19	20	21
Doğruluk	90	90	90	90	90	90	90
İterasyon	22	23	24	25			
Doğruluk	90	90	90	90			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.20).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	8 (DP)	7 (YP)
	N	1 (YN)	64 (DN)

Şekil 6.20. SPCTF veri kümesinde KA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{8 + 64}{8 + 7 + 64 + 1} = 0,9$$

$$\text{Hata oranı} = \frac{7 + 1}{9 + 6 + 64 + 1} = 0,1$$

$$\text{Doğru pozitif oranı} = \frac{8}{8 + 1} = 0,8888$$

$$\text{Yanlış pozitif oranı} = \frac{7}{7 + 64} = 0,0985$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,8888$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,0985 = 0,9015$$

$$\text{hassasiyet} = \frac{8}{8 + 1} = 0,888$$

$$\text{yeniden çağırma} = \frac{DP}{P}$$

$$\text{yeniden çağırma} = \frac{8}{9} = 0,888$$

$$\text{Doğruluk} = \frac{DP + DN}{P + N}$$

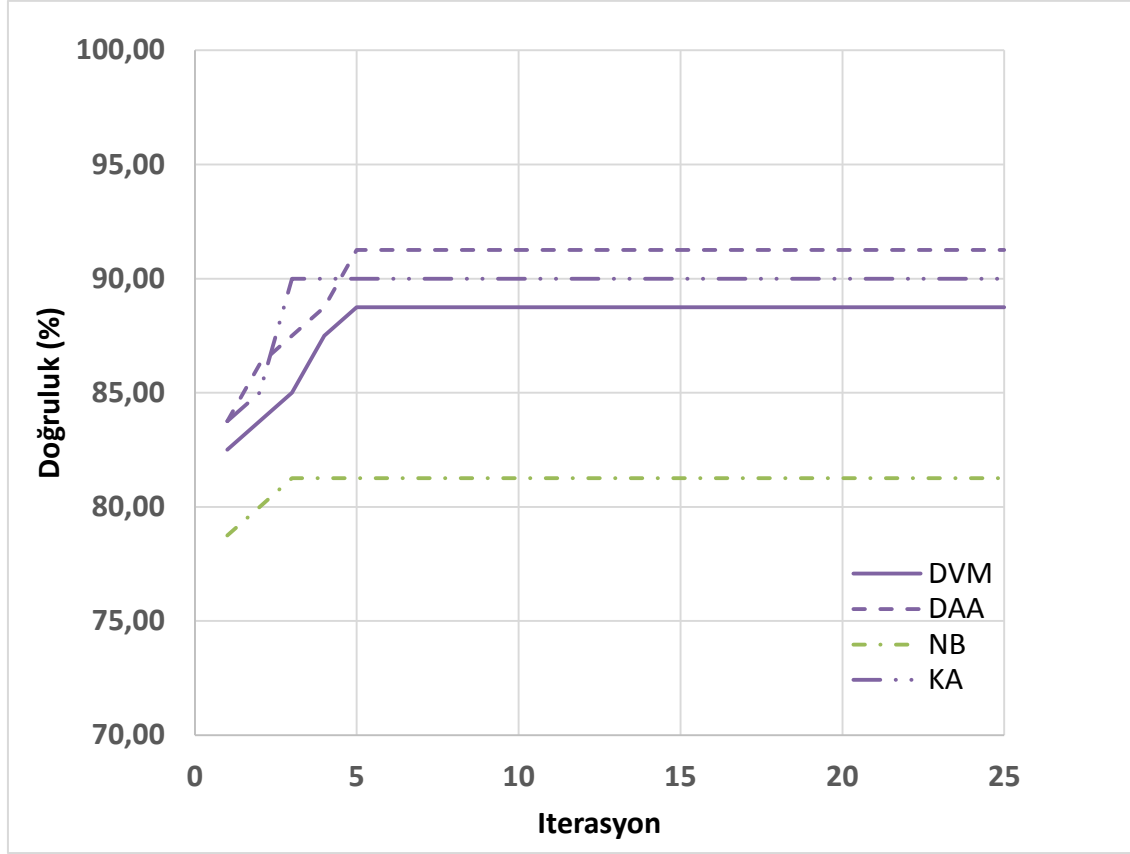
$$\text{Doğruluk} = \frac{8 + 64}{15 + 65} = 0,9$$

$$F - \text{ölçü} = \frac{2}{1/0,888 + 1/0,888} = 0,888$$

Çizelge 6.31’de, DAA 9 öznitelik ile {3, 7, 27, 31, 32, 33, 36, 44, 45} en yüksek doğruluğu 91,25% elde eden sınıflandırıcıdır. Sonra KA 5 öznitelik {31, 32, 33, 42, 43} ve 90% başarı ile yer almaktadır. DVM, 10 öznitelik {3, 11, 12, 24, 32, 33, 39, 41, 44, 45} ile 88,75% doğruluğu elde ederek üçüncü sırada yer alır. NB sadece bir öznitelik {36} ile 81,25% doğruluk ile en düşük sonucu gösterir. Şekil 6.21, 25 iterasyon için DVM, NB, KA ve DAA sınıflandırılma doğruluklarını göstermektedir.

Çizelge 6.31. SPECTF veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi	Doğruluk(%)
DVM	10	3, 11, 12, 24, 32, 33, 39, 41, 44, 45	88,7500
DAA	9	3, 7, 27, 31, 32, 33, 36, 44, 45	91,2500
NB	1	36	81,2500
KA	5	31, 32, 33, 42, 43	90



Şekil 6.21. SPECTF veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları

6.3.3. Satlog Kalp veri kümesi

DVM

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, DVM'nin doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Her tekrarlama her bireyin sınıflandırma doğruluğu hesaplanır, Çizelge 6.32'de her iterasyonda sınıflandırma doğruluğu en yüksek olan bireyin doğruluğunu göstermektedir. Çizelge 6.32'de görebileceğimiz gibi, DVM ilk tekrarda 87,6543% doğruluk ile başlar, sonra ikinci iterasyonda 88,8889% çoğalır, üçüncü iterasyonda 91,358% elde eder, dördüncü iterasyonda 95,0617% sonuçlar, beşinci iterasyonda 96,2963% doğruluğa ulaşır ve bu doğruluk ile 25'inci iterasyona kadar devam eder. Beşinci iterasyonda, DVM, 4 öznelikten oluşan {3, 9, 12, 13} alt kümesi ile 96,2963% sınıflandırma doğruluğu elde eder.

Çizelge 6.32. Statlog veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	87,6543	88,8889	91,358	95,0617	96,2963	96,2963	96,2963
İterasyon	8	9	10	11	12	13	14
Doğruluk	96,2963	96,2963	96,2963	96,2963	96,2963	96,2963	96,2963
İterasyon	15	16	17	18	19	20	21
Doğruluk	96,2963	96,2963	96,2963	96,2963	96,2963	96,2963	96,2963
İterasyon	22	23	24	25			
Doğruluk	96,2963	96,2963	96,2963	96,2963			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.22).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	40 (DP)	0 (YP)
	N	3 (YN)	38 (DN)

Şekil 6.22. Statlog veri kümesinde DVM sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{40 + 38}{40 + 0 + 38 + 3} = 0,9629$$

$$\text{Hata oranı} = \frac{0 + 3}{40 + 0 + 38 + 3} = 0,0370$$

$$\text{Doğru pozitif oranı} = \frac{40}{40 + 3} = 0,9302$$

$$\text{Yanlış pozitif oranı} = \frac{0}{0 + 38} = 0$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,9302$$

$$\text{Seicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0 = 1$$

$$\text{hassasiyet} = \frac{40}{40 + 3} = 0,9302$$

$$\text{yeniden ağırma} = \frac{40}{43} = 0,9302$$

$$\text{Doğruluk} = \frac{40 + 38}{40 + 41} = 0,9629$$

$$F - \text{ölü} = \frac{2}{1/0,9302 + 1/0,9302} = 0,9302$$

DAA

GA algoritmasını alıřtırdığımızda uygunluk fonksiyonunda, DAA'nın doğruluğunu kullandığımızda, ařağıdaki sonuçlar elde edilmektedir.

izelge 6.33'de DAA ilk iterasyonda 86,4198% doğruluk ile başlar ve ikinci iterasyonda 92,5926% yükselir, 3'üncü iterasyonda 93,8272% doğrulukları elde etmektedir. 3'üncü iterasyonda 3 öznitelikten oluřan {9, 12, 13} alt kümesi ile 93,8272% sınıflandırma doğruluğuna ulaşır.

Çizelge 6.33. Statlog veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	86,4198	92,5926	93,8272	93,8272	93,8272	93,8272	93,8272
İterasyon	8	9	10	11	12	13	14
Doğruluk	93,8272	93,8272	93,8272	93,8272	93,8272	93,8272	93,8272
İterasyon	15	16	17	18	19	20	21
Doğruluk	93,8272	93,8272	93,8272	93,8272	93,8272	93,8272	93,8272
İterasyon	22	23	24	25			
Doğruluk	93,8272	93,8272	93,8272	93,8272			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.23).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	38 (DP)	2 (YP)
	N	3 (YN)	38 (DN)

Şekil 6.23. Statlog veri kümesinde DAA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{38 + 38}{38 + 2 + 38 + 3} = 0,9382$$

$$\text{Hata oranı} = \frac{2 + 3}{40 + 0 + 38 + 3} = 0,0617$$

$$\text{Doğru pozitif oranı} = \frac{38}{38 + 3} = 0,9268$$

$$\text{Yanlış pozitif oranı} = \frac{2}{2 + 38} = 0,05$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,9268$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,05 = 0,95$$

$$\text{hassasiyet} = \frac{38}{38 + 3} = 0,9268$$

$$\text{yeniden çağırma} = \frac{38}{41} = 0,9268$$

$$\text{Doğruluk} = \frac{38 + 38}{40 + 41} = 0,9382$$

$$F - \text{ölçü} = \frac{2}{1/0,9268 + 1/0,9268} = 0,9268$$

NB

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, NB'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.34'de görüldüğü gibi, NB yöntemi ilk tekrarda 88,8889% sınıflandırma doğruluk ile başlar ve yükselmeye devam eder böyle ki ikinci iterasyonda 90,1235% doğruluğa yükselir, üçüncü iterasyonda 91,358% ve dördüncü iterasyonda 95,0617% doğruluğu elde eder. NB, dördüncü iterasyonda 3 öznitelik {9, 12, 13} ile 95,0617% doğruluğa ulaşır.

Çizelge 6.34. Statlog veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	88,8889	90,1235	91,358	95,0617	95,0617	95,0617	95,0617
İterasyon	8	9	10	11	12	13	14
Doğruluk	95,0617	95,0617	95,0617	95,0617	95,0617	95,0617	95,0617
İterasyon	15	16	17	18	19	20	21
Doğruluk	95,0617	95,0617	95,0617	95,0617	95,0617	95,0617	95,0617
İterasyon	22	23	24	25			
Doğruluk	95,0617	95,0617	95,0617	95,0617			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.24).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	39 (DP)	1 (YP)
	N	3 (YN)	38 (DN)

Şekil 6.24. Statlog veri kümesinde NB sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{39 + 38}{39 + 1 + 38 + 3} = 0,9506$$

$$\text{Hata oranı} = \frac{2 + 3}{40 + 0 + 38 + 3} = 0,0493$$

$$\text{Doğru pozitif oranı} = \frac{39}{39 + 3} = 0,9285$$

$$\text{Yanlış pozitif oranı} = \frac{1}{1 + 38} = 0,0256$$

Duyarlılık (Sensitivity) = $DP \text{ oranı} = 0,9285$

Seçicilik (Specifity) = $1 - YP \text{ oranı} = 1 - 0,0256 = 0,9744$

hassasiyet = $\frac{39}{39 + 3} = 0,8571$

yeniden çağırma = $\frac{39}{42} = 0,8571$

Doğruluk = $\frac{39 + 38}{40 + 41} = 0,9506$

F – ölçü = $\frac{2}{1/0,8571 + 1/0,8571} = 0,8571$

Karar Ağacı

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, KA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.35'de KA, ilk iterasyonda 87,6543% doğruluğu gösterir, ikinci iterasyonda 91,358%, üçüncü iterasyonda 92,5926% doğruluğa ulaşır. KA, üçüncü iterasyonda 5 öznitelik {3, 9, 11, 12, 13} ile 92,5926% sınıflandırma doğruluğuna ulaşır ve 25'inci itersyona kadar bu doğrulukla devam eder.

Çizelge 6.35. Statlog veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	87,6543	91,358	92,5926	92,5926	92,5926	92,5926	92,5926
İterasyon	8	9	10	11	12	13	14
Doğruluk	92,5926	92,5926	92,5926	92,5926	92,5926	92,5926	92,5926
İterasyon	15	16	17	18	19	20	21
Doğruluk	92,5926	92,5926	92,5926	92,5926	92,5926	92,5926	92,5926
İterasyon	22	23	24	25			
Doğruluk	92,5926	92,5926	92,5926	92,5926			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.25).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	40 (DP)	0 (YP)
	N	6 (YN)	35 (DN)

Şekil 6.25. Statlog veri kümesinde KA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{40 + 35}{40 + 0 + 35 + 6} = 0,9259$$

$$\text{Hata oranı} = \frac{0 + 6}{40 + 0 + 35 + 6} = 0,0740$$

$$\text{Doğru pozitif oranı} = \frac{40}{40 + 6} = 0,8695$$

$$\text{Yanlış pozitif oranı} = \frac{0}{0 + 35} = 0$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,8695$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0 = 1$$

$$\text{hassasiyet} = \frac{40}{40 + 6} = 0,8695$$

$$\text{yeniden çağırma} = \frac{40}{46} = 0,8695$$

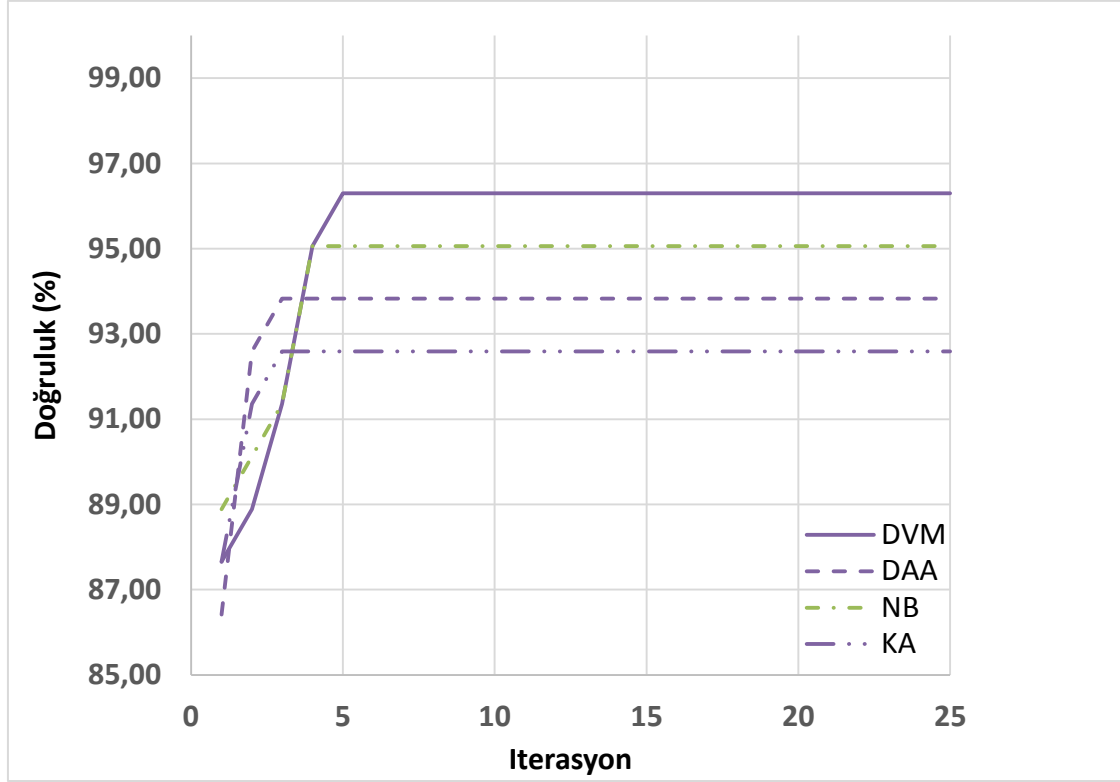
$$\text{Doğruluk} = \frac{40 + 35}{40 + 41} = 0,9259$$

$$F - \text{ölçü} = \frac{2}{1/0,8695 + 1/0,8695} = 1$$

Çizelge 6.36’de DVM, 4 öznitelik {3, 9, 12, 13} ile 96,2963% doğrulukla diğerlerinden daha başarılı olduğunu göstermektedir. Sonra, NB 3 öznitelik {9, 12, 13} ile 95,0617% başarıyı elde etmiştir. DAA aynı öznitelik kümesi ile 93,8272% doğruluğu sonuçlanmıştır. KA, {3, 9, 11, 12, 13} öznitelik alt kümesi ve 92,5926% doğruluk ile en düşük başarıyı elde etmektedir. Şekil 6.26, 25 iterasyon için DVM, NB, KA ve DAA sınıflandırılma doğruluklarını göstermektedir.

Çizelge 6.36. Statlog veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi	Doğruluk(%)
DVM	4	3, 9, 12, 13	96,2963
DAA	3	9, 12, 13	93,8272
NB	3	9, 12, 13	95,0617
KA	5	3, 9, 11, 12, 13	92,5926



Şekil 6.26. Statlog veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları

6.3.4. WPBC veri kümesi

DVM

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, DVM'nin doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Her tekrarlama her bireyin sınıflandırma doğruluğu hesaplanır, Çizelge 6.37'de her iterasyonda sınıflandırma doğruluğu en yüksek olan bireyin doğruluğunu göstermektedir. Çizelge 6.37'de görebileceğimiz gibi, DVM ilk tekrarda 79,3103% doğruluk ile başlar, sonra ikinci iterasyonda 81,0345% çoğalır, üçüncü iterasyonda 82,7586% elde eder, dördüncü, beşinci ve altıncı iterasyonlarda sırayla 84,4828%, 86,2069% ve 87,931% doğrulukları sonuçlar. Yedinci iterasyonda 89,6552% doğruluğa ulaşır ve bu doğruluk ile 25'inci iterasyona kadar devam eder. Yedinci iterasyonda, DVM, 4 öznelikten oluşan {5, 11, 16, 34, 35} alt kümesi ile 89,6552% sınıflandırma doğruluğu elde eder.

Çizelge 6.37. WPBC veri kümesinde, 25 iterasyona dayalı DVM sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	79,3103	81,0345	82,7586	84,4828	86,2069	87,931	89,6552
İterasyon	8	9	10	11	12	13	14
Doğruluk	89,6552	89,6552	89,6552	89,6552	89,6552	89,6552	89,6552
İterasyon	15	16	17	18	19	20	21
Doğruluk	89,6552	89,6552	89,6552	89,6552	89,6552	89,6552	89,6552
İterasyon	22	23	24	25			
Doğruluk	89,6552	89,6552	89,6552	89,6552			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.27).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	7 (DP)	1 (YP)
	N	5 (YN)	45 (DN)

Şekil 6.27. WPBC veri kümesinde DVM sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{7 + 45}{7 + 1 + 45 + 5} = 0,8965$$

$$\text{Hata oranı} = \frac{1 + 5}{7 + 1 + 45 + 5} = 0,1034$$

$$\text{Doğru pozitif oranı} = \frac{7}{7 + 5} = 0,5833$$

$$\text{Yanlış pozitif oranı} = \frac{1}{1 + 45} = 0,0217$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,5833$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,0217 = 0,9783$$

$$\text{hassasiyet} = \frac{7}{7 + 5} = 0,5833$$

$$\text{yeniden çağırma} = \frac{7}{12} = 0,5833$$

$$\text{Doğruluk} = \frac{7 + 45}{8 + 50} = 0,8965$$

$$F - \text{ölçü} = \frac{2}{1/0,5833 + 1/0,5833} = 0,5833$$

DAA

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, DAA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.38'de DAA ilk iterasyonda 87,931% doğruluk ile başlar ve ikinci iterasyonda 89,6552% yükselir, 3'üncü ve 4'üncü iterasyonda sırayla 91,3793% ve 93,1034% doğrulukları elde etmektedir. 5'inci iterasyonda 3 öznitelikten oluşan {3, 13, 24} alt kümesi ile 94,8276% sınıflandırma doğruluğuna ulaşır.

Çizelge 6.38.WPBC veri kümesinde, 25 iterasyona dayalı DAA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	87,931	89,6552	91,3793	93,1034	94,8276	94,8276	94,8276
İterasyon	8	9	10	11	12	13	14
Doğruluk	94,8276	94,8276	94,8276	94,8276	94,8276	94,8276	94,8276
İterasyon	15	16	17	18	19	20	21
Doğruluk	94,8276	94,8276	94,8276	94,8276	94,8276	94,8276	94,8276
İterasyon	22	23	24	25			
Doğruluk	94,8276	94,8276	94,8276	94,8276			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.28).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	5 (DP)	3 (YP)
	N	0 (YN)	50 (DN)

Şekil 6.28. WPBC veri kümesinde DAA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{5 + 50}{5 + 3 + 50 + 0} = 0,9482$$

$$\text{Hata oranı} = \frac{YP + YN}{DP + YP + DN + YN}$$

$$\text{Hata oranı} = \frac{3 + 0}{5 + 3 + 50 + 0} = 0,0517$$

$$\text{Doğru pozitif oranı} = \frac{DP}{DP + YN}$$

$$\text{Doğru pozitif oranı} = \frac{5}{5 + 0} = 1$$

$$\text{Yanlış pozitif oranı} = \frac{YP}{YP + DN}$$

$$\text{Yanlış pozitif oranı} = \frac{3}{3 + 50} = 0,0566$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 1$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,0566 = 0,9434$$

$$\text{hassasiyet} = \frac{5}{5 + 0} = 1$$

$$\text{yeniden çağırma} = \frac{5}{5} = 1$$

$$\text{Doğruluk} = \frac{5 + 50}{8 + 50} = 0,9482$$

$$F - \text{ölçü} = \frac{2}{1/1 + 1/1} = 1$$

NB

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, NB'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.39'de görüldüğü gibi, NB yöntemi ilk tekrarda 76,4706% sınıflandırma doğruluk ile başlar ve yükselmeye devam eder böyle ki ikinci iterasyonda 77,2059% yükselir, üçüncü, dördüncü ve beşinci iterasyonda sırasıyla 78,6765%, 89,6552% ve 91,3793% doğrulukları elde eder. NB, altıncı iterasyonda 3 öznitelik {3, 11, 14} ile 93,1034% doğruluğa ulaşır.

Çizelge 6.39. WPBC veri kümesinde, 25 iterasyona dayalı NB sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	76,4706	77,2059	78,6765	89,6552	91,3793	93,1034	93,1034
İterasyon	8	9	10	11	12	13	14
Doğruluk	93,1034	93,1034	93,1034	93,1034	93,1034	93,1034	93,1034
İterasyon	15	16	17	18	19	20	21
Doğruluk	93,1034	93,1034	93,1034	93,1034	93,1034	93,1034	93,1034
İterasyon	22	23	24	25			
Doğruluk	93,1034	93,1034	93,1034	93,1034			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.29).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	6 (DP)	2 (YP)
	N	2 (YN)	48 (DN)

Şekil 6.29. WPBC veri kümesinde NB sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{6 + 48}{6 + 2 + 48 + 2} = 0,9310$$

$$\text{Hata oranı} = \frac{2 + 2}{6 + 2 + 48 + 2} = 0,0689$$

$$\text{Doğru pozitif oranı} = \frac{6}{6 + 2} = 0,75$$

$$\text{Yanlış pozitif oranı} = \frac{2}{2 + 48} = 0,04$$

$$\text{Duyarlılık (Sensitivity)} = DP \text{ oranı} = 0,75$$

$$\text{Seçicilik (Specifity)} = 1 - YP \text{ oranı} = 1 - 0,04 = 0,96$$

$$\text{hassasiyet} = \frac{6}{6 + 2} = 0,75$$

$$\text{yeniden çağırma} = \frac{6}{8} = 0,75$$

$$\text{Doğruluk} = \frac{6 + 48}{8 + 50} = 0,9310$$

$$F - \text{ölçü} = \frac{2}{1/0,75 + 1/0,75} = 0,7501$$

Karar Ağacı

GA algoritmasını çalıştırdığımızda uygunluk fonksiyonunda, KA'nın doğruluğunu kullandığımızda, aşağıdaki sonuçlar elde edilmektedir.

Çizelge 6.40'de KA, ilk iterasyonda 81,0345% doğruluğu gösterir, ikinci iterasyonda 82,7586%, üçüncü iterasyonda 84,4828% doğruluğa ulaşır. KA, dördüncü iterasyonda 4 öznitelik {11, 14, 16, 35} ile 86,2069% sınıflandırma doğruluğuna ulaşır ve 25'inci itasyona kadar bu doğrulukla devam eder.

Çizelge 6.40. WPBC veri kümesinde, 25 iterasyona dayalı KA sınıflandırılma doğruluğu

İterasyon	1	2	3	4	5	6	7
Doğruluk	81,0345	82,7586	84,4828	86,2069	86,2069	86,2069	86,2069
İterasyon	8	9	10	11	12	13	14
Doğruluk	86,2069	86,2069	86,2069	86,2069	86,2069	86,2069	86,2069
İterasyon	15	16	17	18	19	20	21
Doğruluk	86,2069	86,2069	86,2069	86,2069	86,2069	86,2069	86,2069
İterasyon	22	23	24	25			
Doğruluk	86,2069	86,2069	86,2069	86,2069			

Aşağıda hata matrisi ve ondan hesaplanan ortak performans ölçütleri gösterilmiştir (Şekil 6.30).

		Gerçek sınıf	
		p	n
Tahmin edilen sınıf	P	3 (DP)	5 (YP)
	N	3 (YN)	47 (DN)

Şekil 6.30 WPBC veri kümesinde KA sınıflandırıcının hata matrisi

$$\text{Sınıflandırma doğruluğu} = \frac{3 + 47}{3 + 5 + 47 + 3} = 0,8620$$

$$\text{Hata oranı} = \frac{5 + 3}{3 + 5 + 47 + 3} = 0,1379$$

$$\text{Doğru pozitif oranı} = \frac{3}{3 + 3} = 0,5$$

$$\text{Yanlış pozitif oranı} = \frac{5}{5 + 47} = 0,0961$$

Duyarlılık (Sensitivity) = $DP \text{ oranı} = 0,5$

Seçicilik (Specifity) = $1 - YP \text{ oranı} = 1 - 0,0961 = 0,9039$

$$\text{hassasiyet} = \frac{3}{3 + 3} = 0,5$$

$$\text{yeniden çağırma} = \frac{3}{6} = 0,5$$

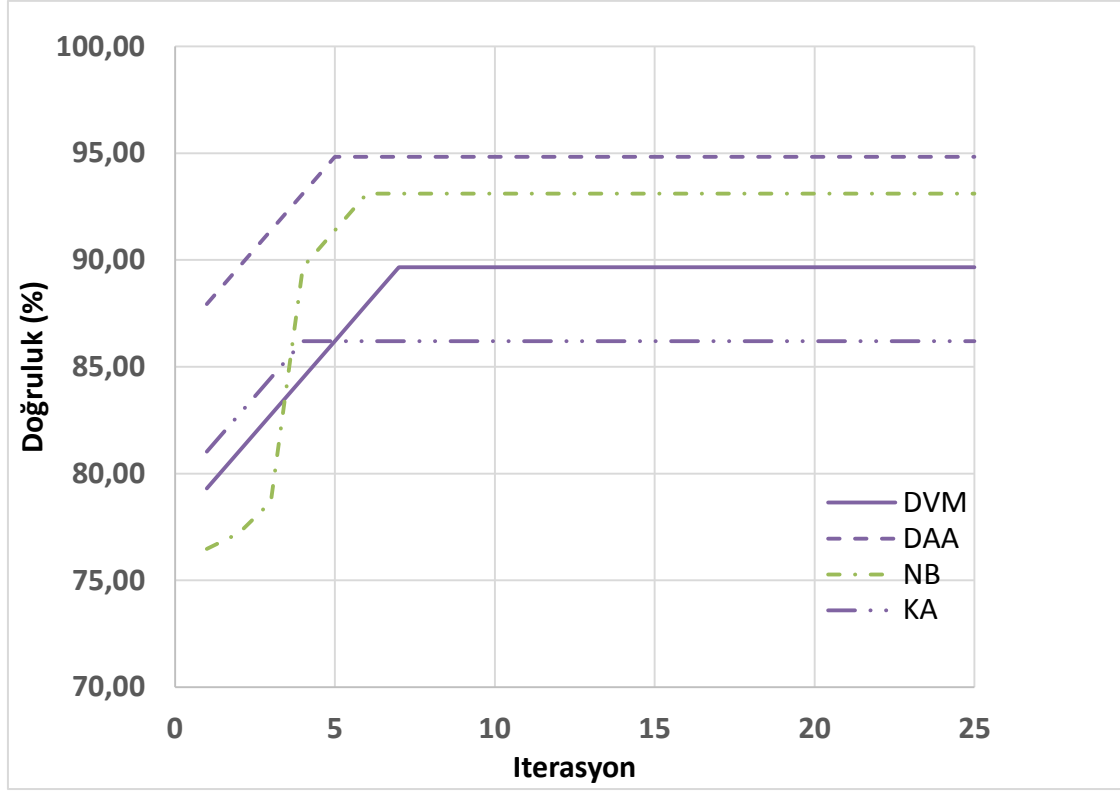
$$\text{Doğruluk} = \frac{3 + 47}{8 + 50} = 0,8620$$

$$F - \text{ölçü} = \frac{2}{1/0,5 + 1/0,5} = 0,5$$

Çizelge 6.41’de görüldüğü gibi DAA, {3, 13, 24} öznitelik kümesi için 94,8276% doğruluk oranı ile diğer yöntemlerden daha başarılıdır. Sonra NB, {3, 11, 14} öznitelik alt kümesi için 93,1034% doğruluğu elde etmektedir. DVM, 5 öznitelik {5, 11, 16, 34, 35} ile 89,6552% doğruluğu sonuçlar ve üçüncü sırada yer alır. KA yöntemi, 4 öznitelik {11, 14, 16, 35} ve 86,2069% başarı ile en düşük doğruluğu elde ediyor. Şekil 6.31, 25 iterasyon için DVM, NB, KA ve DAA sınıflandırılma doğruluklarını göstermektedir.

Çizelge 6.41. WPBC veri kümesinde, DVM, DAA, NB ve KA sınıflandırma yöntemlerin doğrulukları ve seçilen özniteliklerin alt kümesi

Yöntem	Öznitelik sayısı	Öznitelik alt kümesi	Doğruluk(%)
DVM	5	5, 11, 16, 34, 35	89,6552
DAA	3	3, 13, 24	94,8276
NB	3	3, 11, 14	93,1034
KA	4	11, 14, 16, 35	86,2069



Şekil 6.31. WPBC veri kümesinde, 25 iterasyona dayalı DVM, NB, KA ve DAA sınıflandırılma doğrulukları

7. SONUÇ VE ÖNERİLER

Sınıflandırma başarısını arttırmak araştırmacılar tarafından çok sık çalışılan bir konudur. Etkin bir sınıflandırma için algoritmaların başarısı önemlidir. Başarıyı etkileyen diğer faktör veri kümelerinin sahip olduğu niteliklerdir. İlgisiz ya da gürültü olarak adlandırılan nitelikler sınıflandırma başarısını olumsuz etkilemektedir.

Etkin sınıflandırma için ilgisiz niteliklerin atılması ya da veri kümesini en iyi tanımlayan niteliklerin yani özniteliklerin bulunması önemli bir çalışma konusudur. Gerçekleştirilen bu çalışmada etkin, güçlü bir hibrit öznitelik seçme modeli önerilmiştir. Önerilen hibrit metot iki aşamadan oluşmaktadır. İlk aşamada, FDR ve Relief filtreleme yöntemleri ile filtreleme yapılarak bir öznitelik havuzu oluşturulmuştur. Gereksiz veya ilgisiz nitelikler atılmış ve sonuç olarak sonraki aşama için öznitelik havuzu oluşturulmuştur. İkinci aşamada, GA, yüksek doğruluğa ve küçük boyuta sahip olan en uygun öznitelik alt kümesini bulmaya çalışır. GA’da öznitelik alt kümelerini değerlendirmede dört farklı sınıflandırma yöntemi (DVM, DAA, NB ve KA) denenmiştir. Bu şekilde GA algoritması her sınıflandırma yöntemi için tek tek uygulanmıştır. Önerilen metodun başarısı, UCI veri tabanından alınan dört veri kümesi (WDBC, SPECTF, Statlog ve WPBC) üzerinde test edilmiştir.

Meme kanseri veri kümesi üzerinde herhangi bir öznitelik seçimi yöntemi kullanmadan (orijinal 30 nitelik için) ve önerilen yöntemi kullanılarak, DVM, DAA, NB ve KA yöntemlerin öznitelik seçimi doğruluğu Çizelge 7.1’de verilmiş ve elde edilen sonuçlar ile karşılaştırılmıştır. Önerilen GA tabanlı hibrit öznitelik seçimi yöntemi uygulanarak meme kanserini sınıflandırmak için önemli bir performans artışı sağlandığı gösterilmiştir. Dört ayrı sınıflandırıcı için DVM’nin 100% doğruluk ile diğerlerinden daha başarılı olduğu görülmüştür.

Çizelge 7.1. WDBC veri kümesinin Orijinal 30 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması

Yöntem	Doğruluk (%) tüm nitelikler	Doğruluk(%) seçilen öznitelikler
DVM	93,5294	100
DAA	92,9412	99,4118
NB	90,5882	99,4118
KA	88,8235	97,0588

SPECTF veri kümesinin tüm öznitelikleri ve önerilen yaklaşım tarafından seçilen öznitelikler için dört sınıflandırma yönteminin doğrulukları Çizelge 7.2’de gösterilmektedir. Seçilen öznitelikler ile elde edilen doğruluklar, tüm sınıflandırıcılar için artış göstermiştir. Hibrit yaklaşımın daha iyi sonuçlar elde ettiği görülmektedir. Dört sınıflandırıcının sınıflandırma doğruluğu karşılaştırıldığı zaman, DAA 91,25% doğruluk ile diğerlerinden daha başarılı sonuçlar gösterir.

Çizelge 7.2. SPECTF veri kümesinin orijinal 44 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması

Yöntem	Doğruluk (%) tüm nitelikler	Doğruluk(%) seçilen öznitelikler
DVM	73,7500	88,7500
DAA	71,2500	91,2500
NB	70	81,2500
KA	76,2500	90

Statlog veri kümesi üzerinde önerilen yöntemi ile elde edilen öznitelikleri ve ayrıca tüm öznitelikleri kullanarak, dört sınıflandırma yönteminin doğrulukları Çizelge 7.3’de gösterilmektedir. Burada, önerilen yöntemin dikkat çeken bir artışa sebep olduğunu görebiliriz. Dört sınıflandırıcı yöntemlerin tüm sınıflandırma doğruluğu karşılaştırıldığı zaman, DVM 96,26% doğruluk ile diğerlerinden daha başarılı sonuçlar gösterir.

Çizelge 7.3. Statlog veri kümesinin orijinal 13 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması

Yöntem	Doğruluk (%) tüm nitelikler	Doğruluk(%) seçilen öznitelikler
DVM	88,8889	96,2963
DAA	86,4198	93,8272
NB	92,5926	95,0617
KA	70,3704	92,5926

Çizelge 7.4’de dört sınıflandırma yöntemlerin doğruluk oranını WPBC veri kümesi üzerinde tüm öznitelikler ve seçilen öznitelikler için göstermektedir. İki denemeden elde edilen sonuçları karşılaştırdığımızda, sonuçlar da artış olduğunu ve sınıflandırma doğruluğunun yükseldiğini görebiliriz. Dört sınıflandırma yönteminin doğrulukları karşılaştırıldığı zaman, DAA ve NB 93,10% doğruluk ile diğerlerinden daha başarılı sonuçlar gösterir.

Çizelge 7.4. WPBC veri kümesinin orijinal 32 nitelik ve ayrıca seçilen öznitelikler için DVM, DAA, NB ve KA'nin sınıflandırma doğruluklarının karşılaştırılması

Yöntem	Doğruluk (%) tüm nitelikler	Doğruluk(%) seçilen öznitelikler
DVM	77,5862	89,6552
DAA	86,2069	94,8276
NB	79,3103	93,1034
KA	70,6897	86,2069

Çizelge 7.5, Çizelge 7.6, Çizelge 7.7 ve Çizelge 7.8’de dört farklı veri küme için önerilen metot ve literatürde yer alan önceki çalışmalarla karşılaştırmaya yer verilmiştir. Önerilen metot UCI’den elde edilen WDBC, SPECTF, Statlog ve WPBC veri kümelerinde denenmiş, sırayla 100%, 91,25%, 96,29%, ve 94,8276% sınıflandırma başarısı ile etkin bir öznitelik seçme gerçekleştirilebilmiştir.

Sonuçlardan da görebileceğimiz gibi, önerilen yöntem, önceki çalışmalara göre oldukça yüksek sınıflandırma doğruluğuna sahiptir. Sonuçlara göre, sınıflandırma doğruluğu,

öznitelik küme boyutu, eğitim ve test küme oranı gibi bazı temel kriterlerin değerlendirilmesine dayalı, yöntemin diğer çalışmalara göre bazı avantajlarının olduğunu çıkarabiliriz. Bu yüzden çoğu önerilen yaklaşımların aksine, önerilen hibrit metod, küçük öznitelik alt kümesi ve yüksek doğruluk ile etkin bir öznitelik seçme yöntemi olabilir.

Çizelge 7.5. WDBC veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması

Yöntem	Veri kümesi	Öznitelik sayısı	Sınıflandırma doğruluğu (%)
DVM (RTF) [89]	WDBC	30	93,726
DVM(quad)-ICA [90]	WDBC	2	94,41
CART-Symmetric uncert [91]	WDBC	8	94,72
CART-Hybrid approach [92]	WDBC	--	95,96
PSO-DVM [93]	WDBC	17	97,37
K-DVM [94]	WDBC	6	97,38
Multisurface Method Tree [95]	WDBC	--	97,00
DVM [96]	WDBC	--	96,77
SVR [97]	WDBC	--	98,76
QKCDAA [98]	WDBC	--	96,98
AGS-GP [99]	WDBC	--	98,6
Filtre_Sarmalama [100]	WDBC	3	96,8
Önerilen metot GA_DVM	WDBC	5	100

Çizelge 7.6. WPBC veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması

Yöntem	Veri kümesi	Öznitelik sayısı	Sınıflandırma doğruluğu (%)
Filtre_Sarmalama [100]	WPBC	4	84,3
ANN [101]	WPBC	--	70,725
SMO+J48-MLP+IBk (Chi-square ve TBA) [102]	WPBC	--	77,3196
Önerilen metot GA_DAA	WPBC	3	94,8276

Çizelge 7.7. Spectf veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması

Yöntem	Veri kümesi	Öznitelik sayısı	Sınıflandırma doğruluğu (%)
Filtrelam ve Sarmalama [100]	Spectf	5	85,6
Önerilen metot GA_DAA	Spectf	9	91,25

Çizelge 7.8. Statlog veri kümesi için önerilen metodun önceki çalışmalarla karşılaştırılması

Yöntem	Veri kümesi	Öznitelik sayısı	Sınıflandırma doğruluğu (%)
GA-DVM [103]	Statlog	5,4+/-1,85	94,25
IG-ANN [104]	Statlog	8	80,99
ANN [105]	Statlog	--	80,2
Önerilen metot GA_DVM	Statlog	4	96,29

KAYNAKLAR

1. Peng, Y., Wu, Z. and Jiang, J. (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43, 15-23.
2. Yıldız, O., Tez, M., Bilge, HŞ., Akcayol, MA. ve Güler, İ. (2012). Meme kanseri sınıflandırması için very füzyonu ve genetic algoritma tabanlı gen seçimi. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 27, 659-668.
3. Huang, D., Chow, T. W. S. (2005). Efficiently searching the important input variables using Bayesian discriminant. *IEEE Trans. on Circuits and Systems-I: Regular Papers*, 52(4), 785–793.
4. Abe, S., Thawonmas, R. and Kobayashi, Y. (1998). Feature selection by analyzing class regions approximated by ellipsoids. *IEEE Trans. On Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 28(2), 282–287.
5. Liu, H., Dougherty, E. R., Dy, J. G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., Parsons, L., Zhao, Z., Yu, L. and Forman, G. (2005). Evolving feature selection. *IEEE Intelligent Systems*, 20(6), 64–76.
6. Wang, H., Khoshgoftaar, T. M. and Napolitano, A. (2012). Software measurement data reduction using ensemble techniques. *Neurocomputing*, 92, 124–132.
7. Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3, 1289–1305.
8. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439), 531–537.
9. Shotton, J., Winn, J., C. Rother, and Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1), 2–2.
10. Wald, R., Khoshgoftaar, T. M., Napolitano, A., and Sumner, C. (2012). Using Twitter content to predict psychopathy. in *11th International Conference on Machine Learning and Applications (ICMLA)*, 2, 394–401.
11. Nguyen, H., Franke, K., and Petrovic, S. (2010). Improving effectiveness of intrusion detection by correlation feature selection. in *International Conference on Availability, Reliability, and Security (ARES '10)*, 17–24.
12. İnternet: Frank, A. Asuncion, A. UCI Machine Learning Repository. *Irvine, CA: University of California, School of Information and Computer Science*. URL: <http://www.webcitation.org/6VEQo8n6e>, Son Erişim Tarihi: 20.12.2014.
13. Kittler, J. (1986). Feature Selection and Extraction. *Handbook of Pattern Recognition and Image Processing*, T.Y. Young and K.S. Fu, eds., 59-83.

14. Siedlecki, W., Sklansky, J. (1988). On Automatic Feature Selection. *Int'l J. Pattern Recognition and Artificial Intelligence*, 2(2), 197-220.
15. Elkharraz, G., Thumfart, S., Akay, A., Eitzinger, C., and HensonGalal, B. (2014). Making tactile textures with predefined affective properties. *IEEE Transactions on Affective Computing*, 5(1), 57-70.
16. Jain, A., Zongker, D. (1997). Feature Selection: Evaluation, Application, and Small Sample Performance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(2), 153-158.
17. Dash, M., Liu H. (1997). Feature Selection for Classification. *Intelligent Data Analysis*, 1(3), 131-156.
18. Ferri, F. J., Pudil, P., Hatef, M. and Kittler, J. (1994). Comparatiand Study of Techniques for Large-Scale Feature Selection. *Pattern Recognition in Practice IV, E.S. Gelsema and L.N. Kanal, eds.*, 403-413.
19. Kudo M., Sklansky J. (2000). Comparison of Algorithms that Select Features for Pattern Recognition. *Pattern Recognition*, 33(1), 25-41.
20. Piramuthu, S. (1998). Evaluating Feature Selection Methods for Learning in Data Mining Applications. *Proc. 31st Ann. Hawaii Int'l Conf. System Science*, 294-301.
21. 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)*, 1314-1321.
22. 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)*, 4, 2605-2608.
23. Liu, Y., Dellaert, F. (1998). A classification based similarity metric for 3D image retrieval. *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 800-805.
24. Lew, M. S., Principles of visual information retrieval. *Springer*, 2001.
25. Puuronen, S., Tsymbal, A. and Skrypnik, I. (2000). Advanced local feature selection in medical diagnostics. *Proc. 13th IEEE Symp. Computer-Based Medical Systems*, 25-30.
26. Holland J. (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. *Uniandrsity of Michigan Press, Ann Arbor, MIT Press, Cambridge*.
27. Siedlecki, W., Sklansky, J. (1989). A note on genetic algorithms for Large-Scale feature selection. *Pattern Recognition Letters*, 10, 335-347.
28. 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.

29. Yang, J. H., Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2), 44-49.
30. Kuncheva, L. I., Jain, L. C. (1999). Nearest neighbor classifier: simultaneous editing and feature selection. *Pattern Recognition Letters*, 20, 1149-1156.
31. 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), 164-171.
32. Davis, L. (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold.
33. Jog, P., Suh, J. and Gucht, D. (1989). The Effect of population size, heuristic crossover and local improvement on a genetic algorithm for the traveling Salesman problem. *Proc. Int'l Conf. Genetic Algorithms*, 110-115.
34. Bui, T.N., Moon, B.R. (1996). Genetic algorithm and graph partitioning. *IEEE Trans. Computers*, 45(7), 841-855.
35. Zheng, X., Julstrom, B.A. and Cheng, W. (1997). Design of vector quantization codebooks using a genetic algorithm. *Proc. IEEE Int'l Conf. Evolutionary Computation*, 525-529.
36. Saeys, Y. (2004). Feature selection for splice site prediction: a new method using EDA-based feature ranking. *BMC Bioinformatics*, 5.
37. Tan, P., Steinbach, M. and Kumar, V. (2005). Introduction to Data Mining. *Addison Wesley, 1st edition*.
38. Molina, L., Belanche, L. and Nebot, A. (2002). Feature selection algorithms: A survey and experimental evaluation. *In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, 306–313.
39. Guyon, I., Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157–1182.
40. Ladha, L., Deepa, T. (2011). Feature selection methods and algorithms. *International Journal on Computer Science and Engineering (IJCSE)*, 3(5).
41. 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).
42. Cagnoni, S., Dobrzeniecki, A., Poli, R. and Yanch, J. (1999). Genetic algorithm based interactive segmentation of 3D medical images, *Image and Vision Computing*, 17(12) 881–895.
43. Bhanu, B., Poggio, T. (Eds) (1994). Special section on machine learning in computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9).

44. Punch, W., Goodman, E. (1993). Further research on feature selection and classification using genetic algorithms, ***Proceedings of the Fifth International Conference on Genetic Algorithms***, 557–564.
45. Matsui, K., Suganami, Y. and Kosugi, Y. (1999). Feature selection by genetic algorithm for MRI segmentation. ***Systems and Computers in Japan***, 30 (7), 69–78.
46. Emmanouilidis, C., Hunter, A., MacIntyre, J., Cox, C. (1999). Multiple-criteria genetic algorithms for feature selection in neuro-fuzzy modeling. ***Proceedings of the International Joint Conference on Neural Networks, Piscataway***, 6, 4387–4392.
47. Vlachos, M. (2010). Dimensionality reduction. ***In Encyclopedia of Machine Learning***, 274–279.
48. Voiry, M., Amarger, V., Bernier, J. and Madani, K. (2009). Ann-based defects' diagnosis of industrial a optical devices. ***In Juan R. RabuÁ sal, Julian Dorado, and Alejandro Pazos, editors, Encyclopedia of Artificial Intelligence***, 131–137.
49. Grall-Maes, E., Beausero, P. (2002). Mutual information-based feature extraction on the time-frequency plane. ***Signal Processing, IEEE Transactions on***, 50 (4), 779–790.
50. Langley, P. (1994). Selection of relevant features in machine learning. ***In Proceedings of the AAAI Fall symposium on relevance***, 140–144.
51. Wang, H., Chen P. (2009). A feature extraction method based on information theory for fault diagnosis of reciprocating machinery. ***Sensors***, 9 (4), 2415–2436.
52. Alpaydin, E. (2004). Introduction to machine learning. Adaptiand computation and machine learning, ***MIT Press***.
53. Liu, H. (2010). Feature selection. ***In Encyclopedia of Machine Learning***, 402–406.
54. 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***, 137–165.
55. Naqvi, G. (2012). A hybrid filter-wrapper approach for feature selection. ***International Master's Thesis, Studies from the Department of Technology at Örebro Uniandrsity***, 0-104.
56. Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support andctor machines. ***Machine Learning***, 46, 389–422.
57. Stoppiglia, H., Dreyfus, G., Dubois, R. and Oussar, Y. (2003). Ranking a random feature for variable and feature selection. ***Journal of Machine Learning Research.***, 3, 1399–1414.
58. Guyon, I., Gunn, S., Nikraandsh, M. and Lotfi, A. (2006). Feature extraction: foundations and applications (Studies in Fuzziness and Soft Computing), ***Springer-Verlag New York, Inc. Secaucus, NJ, USA***, 207.

59. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: Class discoandry and class prediction by gene expression monitoring. **Science**, 286(5439): 531–537.
60. John, G., Kohavi, R. and Pfleger, K. (1994). Irrelevant features and the subset selection problem. 129, 121–129.
61. S  yndberg-madsen, N., Thomsen, C. and Pe  sa, J. (2003). Unsupervised feature subset selection. *In Proceedings of the Workshop on Probabilistic Graphical Models for Classification*, 71–82.
62. 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*, 137–165.
63. Lauer, F., Suen, C., and Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. **Pattern Recognition.**, 40, 1816–1824.
64. Gunn, S. (1998). Support andctor machines for classification and regression.
65. Kira, K., Rendell, L. A. (1992). A practical approach to feature selection. *in Proceedings of the ninth international workshop on Machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.*, 249–256.
66. Saeys, Y., Inza, A. and Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. **Oxford Uniandrsity Press**, 23(19), 2507–2517.
67. Park, H., Kwon, H. C. (2007). Extended Relief algorithms in instance-based feature filtering. **Advanced Language Processing and Web Information Technology, ALPIT 2007. Sixth International Conference on.**, 123-128.
68. Bouaguel, W., Bel Mufti, G. and Limam, M. (2013). A fusion approach based on wrapper and filter feature selection methods using majority vote and feature weighting. **Computer Applications Technology (ICCAT), 2013 International Conference on**, 1–6.
69. Sabzevari, H., Soleymani, M. and Noorbakhsh, E. A comparison between statistical and Data Mining methods for credit scoring in case of imited available data.
70. Bermejo, P., Ossa, L., Gamez, J. and Puerta, J. (2011). A study on different backward feature selection criteria oandr high-dimensional databases. **Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on**, 1300–1305.
71. Pohl, I. (1970). Bi-directional Search. **IBM T.J. Watson Research Center**, 1970-Database searching, 27.
72. İnternet: Genetic Algorithms, University College Cork Ireland, URL: <http://www.webcitation.org/6VERCiLDF>, Son Eriřim Tarihi: 10.12.2014

73. Lin, J., Yu J. (2011). Weighted naïve bayes classification algorithm based on particle swarm optimization. ***Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on***, 444–447.
74. Yi, Z., Ilic, M. D. and Tonguz, O. (2007). Application of support and/or machine classification to enhanced protection relay logic in electric power grids. ***Power Engineering, 2007 Large Engineering Systems Conference on***, 31–38.
75. Christopher J. C. (1998). A tutorial on support and/or machines for pattern recognition, ***Journal Data Mining and Knowledge Discovery***, 2(2), 121–167.
76. Cortes C., Vapnik V. (1995). Support vector networks. ***Machine Learning***, 20, 273–297.
77. Guyon, I., Vapnik, V., Boser, B., Bottou, L. and Solla, S.A. (1992). Structural risk minimization for character recognition. ***Advances in Neural Information Processing Systems***, 4, 471–479.
78. Aizerman, M. A., Braanderman, E. M. and Rozoner, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. ***Automation and Remote Control***, 25, 821–837.
79. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. (1997). Comparing support and/or machines with gaussian kernels to radial basis function classifiers. ***IEEE Trans. Sign. Processing***, 45, 2758–2765.
80. Quinlan, J. R. (1986). Induction of decision tree, ***Machine Learning***, 1, 81–106.
81. Quinlan, J. R. (1993). C4.5: Programs for machine learning. ***Morgan Kaufmann Publishers, San Mateo, CA***.
82. Breiman, L., Friedman, J. Olshen, H. R. A. and Stone, C. J. (1984). Classification and regression trees. ***Statistics probability series, Wadsworth, Belmont***.
83. Shafer, J., Agrawal, R. and Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. ***Morgan Kaufmann***, 544–555.
84. Hussain, H. M., Benkrid, K., Seker, H. and Erdogan, A. T. (2011). FPGA implementation of K-means algorithm for bioinformatics application: An accelerated approach to clustering microarray data. ***in Proc. Of 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), San Diego, CA, US***, 248–255.
85. Hussain, H. M., Benkrid, K., Seker, H. and Erdogan, A. T. (2011). Highly parametrized K-means clustering on FPGAs: Comparative results with GPPs and GPUs. ***in Proc. of 2011 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Cancun, Mexico***, 475–480.
86. Hussain, H. M., Benkrid, K., Seker, H. (2012). An adaptive implementation of a dynamically reconfigurable K-nearest neighbour classifier on FPGA. ***Adaptive Hardware and Systems (AHS), 2012 NASA/ESA Conference on***, 205–212.

87. Golub, G. H., Loan, C. F. V. (1996). Matrix Computations. (3rd ed.). **Baltimore, MD: The Johns Hopkins University Press.**
88. Fawcett, T. (2006). An introduction to ROC analysis. **Pattern Recognition Letters**, 861–874.
89. Krishnan, A. M, Banerjee, R., Chakraborty, S. and Chakraborty, C. (2010). Statistical analysis of mammographic features and its classification using support andtor machine. **Expert Systems with Applications**, 37, 470-478.
90. Mert, A., Kilic, N. and Akan, A. (2011). Breast cancer classification by using support andtor machines with reduced dimension. **53rd International Symposium ELMAR, Croatia**, 37-40.
91. Lavanya, D., Rani, k. (2011). Analysis of feature selection with classification: breast cancer datasets. **Indian Journal of Computer Science and Engineering**, 2, 756-763.
92. Lavanya, D, Rani, K. (2012). Ensemble decision tree classifier or breast cancer data. **International Journal of Information Technology Conandrgence**, 2,17-24.
93. Prasad, Y., Biswas, K. and Jain, C. (2010). Svm classifier based feature selection using ga, aco and pso for sirna design. **In proceeding of: Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China**. 307-314.
94. Zheng, B., Yoon, S. W. and Lam, S. S. (2014). Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support andtor machine algorithms. **Expert Systems with Applications**, 41, 1476–1482.
95. Gorunescu, F., Belciug, S. (2014). Evolutionary strategy to deandlop learning-based decision systems. **Application to breast cancer and liandr fibrosis stadialization. J Biomed Inform**, 49, 112-118.
96. Stoean, R, Stoean, C. (2013). Modeling medical decision making by support andtor machines, explaining by rules of evolutionary algorithms with feature selection. **Expert Systems with Applications**, 40, 2677–2686.
97. Zangooei, M. H, Habibi, J. and Alizadehsani, R. (2014). Disease diagnosis with a hybrid method SVR using NSGA-II. **Neurocomputing**, 136,14–29.
98. Li, J. B., Peng, Y. and Liu, D. (2013). Quasiconformal kernel common locality discriminant analysis with application to breast cancer diagnosis. **Information Sciences**, 223, 256–269.
99. Zhu, Z., Nandi, A. K. and Aslam, M. W. (2013). Adapted geometric semantic genetic programming for diabetes and breast cancer classification. **2013 IEEE International Workshop on Machine learning for signal processing, UK.**, 22–25.
100. Foithong, S., Pinngern, O. and Attachoo, B. (2012). Feature subset selection wrapper based on mutual information and rough sets. **Expert Systems with Applications**, 39, 574–584.

101. Chuneekar, V. N., Ambulgekar, H. P. (2009). Approach of neural network to diagnose breast cancer on three different data set. **2009 International Conference on Advances in Recent Technologies in Communication and Computing**, 893-895.
102. Gouda I. S., Abdelhalim, M. B. and Abd-elghany Zeid, M. (2012). Breast cancer diagnosis on three different datasets, using Multi-Classifiers, **International Journal of Computer and Information Technology**, 1(1), 2277–0764.
103. Huang, C.L. and Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support andctor machines, **Expert Systems with Applications**, 31, 231–240.
104. Khemphila, A., Boonjing, V. (2011). Heart disease classification using neural network and feature selection systems engineering (ICSEng), **2011 21st International Conference on**. 406 – 409.
105. Khemphila, A., Boonjing, V. (2010). Comparing performances of logistic regression, decision trees, and neural networks for classifying heart disease patients. **Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on**, 193–198.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : AFZALI, SHIMA
Uyruğu : T.C.
Doğum tarihi ve yeri : 23.05.1987, Zanja
Medeni hali : Bekâr
Telefon : 0 (531) 010 32 65
Faks :
e-mail : shimaafzali@os.gazi.edu.tr



Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi / B.M.	Devam Ediyor
Lisans	Zanja Azad University/ B.Y.M.B	2009
Lise	Tohid Lisesi	2005

İş Deneyimi

Yıl	Yer	Görev
-----	-----	-------

Yabancı Dil

İngilizce

Yayınlar

-

Hobiler

Yüzme, Gitar, Dans, Tenis



GAZİ GELECEKTİR..