

T.C.  
DOKUZ EYLÜL ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜ  
EKONOMETRİ ANABİLİM DALI  
EKONOMETRİ PROGRAMI  
DOKTORA TEZİ

SINIFLANDIRMA PROBLEMİNDE MAKİNE  
ÖĞRENMESİ ALGORİTMALARININ PERFORMANSINI  
ARTIRMAK İÇİN HİBRİT BİR YÖNTEM ÖNERİSİ

Meryem PULAT

Danışman  
Prof. Dr. İpek DEVECİ KOCAKOÇ

2023

T.C.  
DOKUZ EYLÜL ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜ  
EKONOMETRİ ANABİLİM DALI  
EKONOMETRİ PROGRAMI  
DOKTORA TEZİ

SINIFLANDIRMA PROBLEMİNDE MAKİNE  
ÖĞRENMESİ ALGORİTMALARININ PERFORMANSINI  
ARTIRMAK İÇİN HİBRİT BİR YÖNTEM ÖNERİSİ

Meryem PULAT

Danışman  
Prof. Dr. İpek DEVECİ KOCAKOÇ

İZMİR - 2023

## TEZ ONAY SAYFASI



## YEMİN METNİ

Doktora Tezi olarak sunduđum “Sınıflandırma Probleminde Makine Öğrenmesi Algoritmalarının Performansını Artırmak için Hibrit bir Yöntem Önerisi” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuđunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

Tarih

.../.../2023

Meryem PULAT



## ÖZET

Doktora Tezi

Sınıflandırma Probleminde Makine Öğrenmesi Algoritmalarının Performansını  
Artırmak için Hibrit bir Yöntem Önerisi

Meryem PULAT

Dokuz Eylül Üniversitesi

Sosyal Bilimler Enstitüsü

Ekonometri Anabilim Dalı

Ekonometri Programı

Makine öğrenmesi, sistemin geçmişteki deneyimlerinden elde edilen öğrenmelerini kullanarak bir model oluşturmasına ve gelecekte karşılaşılabilecek durumlar karşısında bir tahminde bulunmasını sağlayan bir yapay zekâ alanıdır. Sınıflandırma, regresyon ve kümeleme problemlerinin çözümünün gerçekleştirilmesi için çok sayıda makine öğrenme algoritması geliştirilmiştir. Yapılan çalışmada sınıflandırma problemi ele alınmıştır.

Tezin amacı, sınıflandırma probleminin çözümünde kullanılan makine öğrenmesi algoritmalarının performansını artırmaktır. Algoritmaların performansı etkileyen en önemli unsur elimizdeki veri setindeki özniteliklerin yeterli sayıda ve gerekli olmasıdır. Bunun için sınıflandırma işlemini gerçekleştirilmeden önce öznitelik seçimi yapılmaktadır. Öznitelik seçimi olarak da filtreleme ve sarmal yöntemleri birleştiren hibrit bir yaklaşım kullanılmıştır. Bu hibrit yaklaşımla, sadece filtreleme yönteminin kullanıldığı duruma nazaran sınıflandırma doğruluğunu iyileştirmekle kalmıyor, aynı zamanda sadece sarmal yöntemler kullanıldığı durumdaki işlem süresi de azaltılabilmektedir. Öznitelik seçimi sonrası genel olarak algoritmaların performansı iyileşmiştir. Algoritmanın performansını geliştirmek için diğer bir denenmiş yaklaşım ise, yığın (stacking) algoritmasında temel ve meta model olarak farklı algoritmalar denenerek daha iyi performansa sahip model bileşimi bulunmaya çalışılmıştır. Temel ve meta model olarak en iyi performansı verebilecek algoritmalara ilişkin genel geçer bir kabul bulunmamaktadır. Çalışmada farklı temel ve meta modeller kullanılarak 15504 model denenmiştir. Literatürde bu kadar kapsamlı bir model denemesi bulunmamaktadır yani bu konuda önemli eksiklik bulunmaktadır.

**Tüm algoritmalar içerisinde en iyi performansa sahip algoritmanın ardışık öznitelik seçimi sonrası yığın algoritması olduğu görülmüştür.**

**Anahtar Kelimeler: Makine öğrenmesi, Topluluk Öğrenmesi, Yığın, Sınıflandırma, Öznitelik Seçimi.**



## **ABSTRACT**

**Doctoral Thesis**

**A Hybrid Method Proposal to Improve Performance of Machine Learning  
Algorithms in Classification Problem**

**Meryem PULAT**

**Dokuz Eylül University**

**Graduate School Of Social Sciences**

**Department of Econometrics**

**Econometrics Program**

**Machine learning is a field of artificial intelligence that allows the system to create a model using its learning from past experiences and make a prediction about the situations it will encounter in the future. Many machine learning algorithms have been developed to solve classification, regression and clustering problems. In the study, the classification problem is discussed.**

**The aim of the thesis is to improve the performance of machine learning algorithms used in solving the classification problem. The most important factor affecting the performance of algorithms is that the features in the data set we have are sufficient and relevant. For this, feature selection was made before performing the classification process. A hybrid approach combining filtering and wrapper methods is used for feature selection. With this hybrid approach, not only can the classification accuracy be improved when using filtering, but also the processing time can be reduced when only wrapper methods are used. Generally, the performance of algorithms has improved after feature Selection. Another tried to approach to improve the performance of the algorithm is to try to find a combination of models with better performance by trying different algorithms as basic and meta models in the stacking algorithm. The performance of the algorithm varies according to the base and meta-models used in the stacking algorithm. There is no general acceptance of algorithms that can give the best performance as a base and meta-model. In the study, 15504 models were tested using different base and meta models. No such exhaustive model experiment exists in the literature, so there is a significant deficiency in this respect.**

**Among all algorithms, the algorithm with the best performance was found to be the stacking algorithm after sequential feature selection.**

**Keywords: Machine Learning, Ensemble Learning, Stacking, Classification, Feture Selection.**



# SINIFLANDIRMA PROBLEMİNDE MAKİNE ÖĞRENMESİ ALGORİTMALARININ PERFORMANSINI ARTIRMAK İÇİN HİBRİT BİR YÖNTEM ÖNERİSİ

## İÇİNDEKİLER

TEZ ONAY SAYFASI	ii
YEMİN METNİ	iii
ÖZET	iv
ABSTRACT	vi
İÇİNDEKİLER	viii
KISALTMALAR	x
TABLOLAR LİSTESİ	xi
ŞEKİLLER LİSTESİ	xii
EKLER LİSTESİ	xiv
GİRİŞ	1

## BİRİNCİ BÖLÜM MAKİNE ÖĞRENMESİ

1.1. MAKİNE ÖĞRENMESİ KAVRAMI	6
1.2. ÖĞRENME STRATEJİLERİ	7
1.2.1. Denetimli Öğrenme	8
1.2.2. Denetimsiz Öğrenme	9
1.2.3. Takviyeli Öğrenme	9
1.3. SINIFLANDIRMA	9
1.4. ÖZİNİTELİK SEÇİM YÖNTEMLERİ	10
1.4.1. Filtreleme Yöntemleri	11
1.4.2. Sarmal Yöntemler	14
1.5. KULLANILAN MAKİNE ÖĞRENMESİ ALGORİTMALARI	18
1.5.1. K-En Yakın Komşu (K- Nearest Neighbors)	18
1.5.2. Naive Bayes	18
1.5.3. Lojistik Regresyon (Logistic Regression)	19
1.5.4. Destek Vektör Makinesi (Support Vector Machine)	20
1.5.5. Yapay Sinir Ağları (Artificial Neural Networks)	23
1.5.6. Karar Ağaçları (Decision Tree)	24

1.5.7. Topluluk Öğrenmesi Yöntemleri (Ensemble Learning Methods)	30
1.5.7.1. Torbalama (Bagging)	31
1.5.7.2. Artırma (Boosting)	33
1.5.7.3. Rotasyon Ormanı (Rotation Forest)	35
1.5.7.4. Yığın – İstifleme (Stacking)	36
1.6. SINIFLANDIRICI PERFORMANS DEĞERLENDİRME ÖLÇÜTLERİ	37

## **İKİNCİ BÖLÜM LİTERATÜR TARAMASI**

2.1. TEZ ÇALIŞMALARININ LİTERATÜR İNCELEMESİ	40
2.1. MAKALELERİN LİTERATÜR İNCELEMESİ	44
2.2. KULLANILAN VERİ SETLERİNİN LİTERATÜR İNCELEMESİ	50

## **ÜÇÜNCÜ BÖLÜM UYGULAMA**

3.1. VERİ SETİ	53
3.2. KODLAMA	54
3.3. ÖNERİLEN YAKLAŞIM	55
3.3.1. Filtreleme ve Sarmal Yöntemleri Birleştiren Hibrit Öznitelik Seçimi	56
3.3.2. Stacking Algoritması Farklı Model Denemeleri	57
3.4. TEMEL ALGORİTMA SONUÇLARI	59
3.4.1. GlaucomaM Veri Setinin Temel Algoritma Sonuçları	59
3.4.2. WDBC Veri Setinin Temel Algoritma Sonuçları	61
3.5. STACKING ALGORİTMASI SONUÇLARI	63
3.5.1. GlaucomaM Veri Setinin Stacking Algoritması Sonuçları	63
3.5.2. WDBC Veri Setinin Stacking Algoritması Sonuçları	64
3.6. ARDIŞIK ÖZİNİTELİK SEÇİMİ SONUÇLARI	66
3.6.1. GlaucomaM Veri Setinin Ardışık Öznitelik Seçimi Sonuçları	66
3.6.1. WDBC Veri Setinin Ardışık Öznitelik Seçimi Sonuçları	86
SONUÇ	108
KAYNAKÇA	110
EKLER	

## KISALTMALAR

<b>KNN</b>	K-En Yakın Komşu (K-Nearest Neighbors)
<b>NB</b>	Naive Bayes
<b>LR</b>	Lojistik Regresyon (Logistic Regression)
<b>SVM</b>	Destek Vektör Makineleri (Support Vector Machine)
<b>ANN</b>	Yapay Sinir Ağları (Artificial Neural Network)
<b>MLP</b>	Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
<b>DT</b>	Karar Ağaçları (Decision Tree)
<b>LDA</b>	Linear Discriminant Analysis (Doğrusal Diskriminant Analizi)
<b>GBM</b>	Gradyan Artırma (Gradient Boosting Machine)
<b>RF</b>	Rasgele Orman (Random Forest)
<b>RoF</b>	Rotasyon Ormanı (Rotation Forest)
<b>IG</b>	Bilgi Kazancı (Information Gain)
<b>GR</b>	Kazanç Oranı (Gain Ratio)
<b>CFS</b>	Korelasyon Tabanlı Öznitelik Seçimi (Correlation based Feature Selection)
<b>SU</b>	Simetrik Belirsizlik Katsayısı (Symmetrical Uncertainty)
<b>RFE</b>	Özyinelemeli Öznitelik Eliminasyonu (Recursive Feature Elimination)
<b>GA</b>	Genetik Algoritma (Genetic Algorithm)
<b>SA</b>	Tavlama Benzetimi (Simulated Annealing)
<b>TP</b>	Gerçek Pozitif (True Positive)
<b>TN</b>	Gerçek Negatif (True Negative)
<b>FP</b>	Yanlış Pozitif (False Positive)
<b>FN</b>	Yanlış Negatif (False Negative)

## TABLolar LİSTESİ

<b>Tablo 1:</b> Karmaşıklık Matrisi (Confusion Matrix)	s.38
<b>Tablo 2:</b> 500 ve Üzeri Atıf Alan Makaleler	s.45
<b>Tablo 3:</b> Makalelerde Kullanılan Diğer Anahtar Kelimeler	s.49
<b>Tablo 4:</b> WDBC Veri Setinin Literatür İncelemesi	s.50
<b>Tablo 5:</b> GlaucomaM Veri Setinin Literatür İncelemesi	s.52
<b>Tablo 6:</b> Breast Cancer Wisconsin (Diagnostic) (WDBC) Veri Seti	s.53
<b>Tablo 7:</b> Kullanılan Ardışık Öznitelik Seçim Yöntemleri	s.57
<b>Tablo 8:</b> GlaucomaM Veri Setinin Karmaşıklık Matrisi	s.59
<b>Tablo 9:</b> GlaucomaM Veri Seti için Makine Öğrenmesi Algoritmalarının Sonuçları	s.60
<b>Tablo 10:</b> WDBC Veri Setinin Karmaşıklık Matrisi	s.61
<b>Tablo 11:</b> WDBC Veri Seti için Makine Öğrenmesi Algoritmalarının Sonuçları	s.62
<b>Tablo 12:</b> GlaucomaM Veri Seti için %95 Üzeri Doğruluk Veren Stacking Algoritması Sonuçları	s.63
<b>Tablo 13:</b> WDBC Veri Seti için %98 Üzeri Doğruluk Veren Stacking Algoritması Sonuçları	s.65
<b>Tablo 14:</b> GlaucomaM Veri Seti için Ardışık Öznitelik Seçimi ile Seçilen Değişken Sayıları ve Hesaplama Zamanı	s.68
<b>Tablo 15:</b> GlaucomaM Veri Seti için Ardışık Öznitelik Seçimi Sonrası Makine Öğrenmesi Algoritmalarının Sonuçları	s.69
<b>Tablo 16:</b> GlaucomaM Veri Seti için Ardışık Öznitelik Seçimi Sonrası %95 Üzeri Doğruluk Veren Stacking Algoritması Sonuçları	s.80
<b>Tablo 17:</b> WDBC Veri Seti için Ardışık Öznitelik Seçimi ile Seçilen Değişken Sayıları ve Hesaplama Zamanı	s.88
<b>Tablo 18:</b> WDBC Veri Seti için Ardışık Öznitelik Seçimi Sonrası Makine Öğrenmesi Algoritmalarının Sonuçları	s.88
<b>Tablo 19:</b> WDBC Veri Seti için Ardışık Öznitelik Seçimi Sonrası %98 Üzeri Doğruluk Veren Stacking Algoritması Sonuçları	s.99
<b>Tablo 20:</b> Üç defa ve Üzeri Seçilen Temel Modeller	s.106
<b>Tablo 21:</b> Veri Setleri için En İyi Sonucu Veren Algoritmalar	s.107

## ŞEKİLLER LİSTESİ

<b>Şekil 1:</b> Öğrenme Stratejileri	s.8
<b>Şekil 2:</b> Genetik Algoritma Akış Diyagramı	s.17
<b>Şekil 3:</b> Lojistik Fonksiyon (Logistic Function)	s.20
<b>Şekil 4:</b> Mavi ve Mor ile Gösterilen İki Sınıf için Pek Çok Olası Hiperdüzlem Arasından Üç Ayırıcı Hiperdüzlem	s.21
<b>Şekil 5:</b> Mavi ve Mor ile Gösterilen İki Sınıfına İlişkin Örnek	s.22
<b>Şekil 6:</b> Mavi ve Mor ile Gösterilen İki Sınıfına İlişkin Örnek	s.22
<b>Şekil 7:</b> Mavi ve Mor ile Gösterilen Sınıfına Ait Örneklerde Farklı çekirdek Fonksiyon Kullanımı	s.22
<b>Şekil 8:</b> Tek ve Çok Katmanlı Sinir Ağları	s.24
<b>Şekil 9:</b> Karar Ağacı Yapısı	s.26
<b>Şekil 10:</b> Bagging Algoritması	s.32
<b>Şekil 11:</b> Rasgele Orman Algoritması	s.33
<b>Şekil 12:</b> Boosting Algoritması	s.34
<b>Şekil 13:</b> Rotasyon Ormanı Algoritması	s.36
<b>Şekil 14:</b> Stacking Algoritması	s.37
<b>Şekil 15:</b> Tezlerin Yıllara göre Dağılımı	s.40
<b>Şekil 16:</b> Enstitülere göre Tezlerin Frekansı	s.41
<b>Şekil 17:</b> Ana Bilim Dallarına göre Tezlerin Frekansı	s.41
<b>Şekil 18:</b> 5 Defa ve Üzeri Kullanılan Yöntemlerin Frekansı	s.42
<b>Şekil 19:</b> 5'ten Az Kullanılan Yöntemlerin Frekansı	s.42
<b>Şekil 20:</b> Tezlerde Kullanılan Öznitelik Seçim Yöntemlerinin Frekansı	s.43
<b>Şekil 21:</b> Tezlerde Kullanılan Anahtar Kelimelerin Kelime Bulutu	s.44
<b>Şekil 22:</b> Makale Sayılarının Yıllara Göre Dağılımı	s.44
<b>Şekil 23:</b> 5 Defa ve Üzeri Kullanılan Yöntemlerin Frekansı	s.46
<b>Şekil 24:</b> 5'ten Az Kullanılan Yöntemlerin Frekansı	s.47
<b>Şekil 25:</b> Makalelerde 2' den Fazla Kullanılan Öznitelik Seçim Yöntemlerin Frekansı	s.47
<b>Şekil 26:</b> Makalelerde Öznitelik Seçimi ile Birlikte Kullanılan Makine Öğrenme Algoritmalarının Frekansı	s.48
<b>Şekil 27:</b> Makalelerde Sık Kullanılan Anahtar Kelimeler ile Kelime bulutu	s.49
<b>Şekil 28:</b> Önerilen Öznitelik Seçim Yaklaşımının Akış Diyagramı	s.57
<b>Şekil 29:</b> Kullanılan Stacking Algoritması	s.58

<b>Şekil 30:</b> GlaucomaM Veri Seti için Değişkenlerin Seçilme Sayısı	s.67
<b>Şekil 31:</b> GlaucomaM Veri Seti için Değişkenlerin Seçilme Sayısı	s.67
<b>Şekil 32:</b> WDBC Veri Seti için Değişkenlerin Seçilme Sayısı	s.87
<b>Şekil 33:</b> Ardışık Öznitelik Seçim Yöntemlerinin Seçilme Sıklığı	s.104
<b>Şekil 34:</b> Üç defa ve Üzeri Seçilen Temel Modeller	s.105
<b>Şekil 35:</b> Meta Modellerin Seçilme Sıklığı	s.106



## EKLER LİSTESİ

**EK 1:** Stacking Algoritmasında Denenmiş Modeller

ek s.1



## GİRİŞ

Makine öğrenmesi (Machine learning) veriden öğrenmeye ve bilgiyi ortaya çıkarmaya dayalı bir disiplin olarak bilinmektedir. Veride gizli kalmış bilgiyi keşfetmeye yönelik işlemler Veri Bilimi (Data Science) olarak ifade edilmektedir. Veri biliminin temellerine bakacak olursak, 1663'de Londra'daki veba salgınından dolayı ölenlerin oranına ilişkin veriler tutulması olduğu söylenebilir. Modern anlamda veri bilimi ile ilgili ilk gelişme ise, 1887 yılında Herman Hollerith tarafından bulunan ilk bilgisayar ve delikli kartlar yapılan sayım işlemleri akla gelmektedir. Verinin hareketliliği ve gelişimi 70'li yıllarda kişisel bilgisayarların yaygınlaşmasına paralel olarak başlamaktadır. 90'lı yıllarda internet ile veri saklanması ve paylaşılması noktasında büyük gelişme kaydedilmektedir. Son olarak ise, bulut bilişim teknolojisi sayesinde de yine verinin saklanması ve paylaşılmasına büyük katkı sağlamaktadır (Gürsaka, 2019: 19).

Verilerle boğulmuş durumdayız. Dünyadaki ve hayatımızdaki veri miktarı giderek artmaktadır ve bunun bir sonunun olmadığı görülmektedir. Bilgisayarların gelişimi ve neredeyse her yerde var olması daha önce kaydedemediğimiz şeyleri kaydetmemizi çok kolaylaştırmaktadır. Her yerde bulunan cihazlar kararlarımızı, alışverişlerdeki seçimlerimizi, finansal alışkanlıklarımızı vb. kaydetmektedir. Hepimiz, veri üretimi ile onu anlamamız arasındaki büyüyen uçuruma tanıklık etmekteyiz. Veri hacmi arttıkça, insanların anladığı oran, karşı konulamaz bir şekilde, endişe verici bir şekilde azalmaktadır. Tüm bu verilerde potansiyel olarak yararlı bilgiler gizlidir (Witten ve diğerleri, 2005: 3-4).

İnsanlar, insan yaşamı başladığından beri verilerde kalıplar aramaktadır. Avcılar hayvan göçü davranışında kalıplar ararlar, çiftçiler mahsul büyümesinde kalıplar ararlar, politikacılar seçmenlerin görüşünde kalıplar ararlar gibi. Bir bilim insanının işi verileri anlamlandırmak, fiziksel dünyanın nasıl çalıştığını ilişkin kalıpları keşfetmek ve onları yeni durumlarda ne olacağını ilişkin tahmin yapabilmektir. Ekonomistler, istatistikçiler ve iletişim mühendisleri, verilerdeki kalıpların otomatik olarak aranabileceği, tanımlanabileceği, doğrulanabileceği ve tahmin için kullanılabilmesi fikriyle uzun süredir çalışmaktadır. Yeni olan, verilerde kalıp bulma fırsatlarındaki şaşırtıcı artıştır. Son yıllarda veri tabanlarının aşırı büyümesi, müşteri tercihleri gibi günlük faaliyetler için veri tabanları, veri madenciliğini yeni iş teknolojilerinin ön saflarına getirmektedir. Veri akışı arttıkça ve aramayı üstlenebilecek makineler yaygınlaştıkça, veri madenciliği fırsatları artmaktadır.

Akıllıca analiz edilen veriler çok kıymetli bir kaynaktır. Yeni anlayışlara ve ticari ortamlarda rekabet avantajlarına yol açabilir (Witten ve diğerleri, 2005: 3-4).

Ethem Alpaydın kitabında makine öğrenimini "Örnek veriler veya geçmiş deneyimler kullanarak bilgisayarları bir performans kriterini optimize edecek şekilde programlama" alanı olarak tanımlamıştır. Yapılan farklı tanımlamalarda da ortak olarak, "örneklerden öğrenerek bilgisayarların geleneksel sayı hesaplamasının ötesinde görevleri akıllıca gerçekleştirmeleri için yönlendirme fikri" çıkarılabilir. Makine öğrenimi, insan öğrenimi etkinliklerini simüle etmek için bilgisayarların nasıl kullanılacağına ilişkin çalışmadır; bilgisayarların yeni bilgiler ve yeni beceriler kazanması, var olan bilgileri belirlemesi ve performansını sürekli iyileştirmesi ve kendini geliştirme yöntemlerine ulaşması çalışmasıdır. Makine öğrenimi, yapay zekanın alt dallarındandır. Makine öğrenme yeteneğine sahiptir, IBM'in "Deep Blue" makinesi, satranç ustası Garry Kasparov'u yener ve yöntem doğru kullanılırsa, makine üstün bir öğrenme yeteneğine sahip olabilmektedir. İster dama veya satranç oyunları oynayın ister yazılı kalıpları tanıyın ister radyasyon onkolojisindeki büyük sorunları çözün, çevredeki ortamdaki girdi yoluyla öğrenme yeteneği, başarılı bir makine öğrenimi uygulaması geliştirmenin temel anahtarıdır (El Naqa ve Murphy, 2015: 3-5).

Çoğu veri setinde alakalı (relevant), alakasız (irrelevant), ya da gereksiz (redundant) öznitelikler (değişken) bulunmaktadır. Öznitelik seçimi, orijinal N öznitelik kümesinden minimum M öznitelik alt kümesini seçen bir süreç olarak tanımlanabilir, böylece öznitelik uzayı belirli bir değerlendirme kriterine göre en uygun şekilde küçültülür. Özellik seçimi, sınıflandırma algoritmalarının performansında önemli bir rol oynamaktadır. Öznitelik seçimi; algoritmaları hızlandırmak, veri kalitesini ve buna bağlı olarak algoritmaların performansını iyileştirmek ve algoritmaların sonuçlarının anlaşılabilirliğini artırmaktadır. Öznitelik seçim algoritmaları; filtreleme, sarmal ve gömülü yaklaşım olmak üzere üçe ayrılmaktadır (Novaković, 2016: 1). Filtre yöntemleri, esas olarak, öğrenme algoritmalarından bağımsız olarak verilen değerlendirme kriterleri ile orijinal özellik kümesinden bir özellik alt kümesini tanımlar. Sarmal yöntemler ise, belirtilen öğrenme algoritmaları tarafından tahmin edilen yüksek tahmin performansına sahip özellikleri seçmektedir (Xie ve Wang, 2011: 5810). Yapılan çalışmada öznitelikler ilk olarak hesaplama açısından daha hızlı olan filtreleme yöntemleri aracılığıyla orijinal öznitelik kümesinden seçilir. Filtreleme yöntemleri ile seçilen özniteliklere sarmal yöntemler uygulanarak daha da iyileştirilmektedir. Sarmal yöntemlerle nihai öznitelikler elde edilerek sınıflandırma işlemi yapılmaktadır.

Makine öğrenimi sınıflandırma, regresyon ve kümeleme problemlerinin çözümünde kullanılabilir. Yapılan çalışmada sınıflandırma problemleri ele alınmıştır. Sınıflandırma problemlerinde, sınıfları belli olan verileri (eğitim verisi) kullanarak bir model oluşturulur, daha sonra oluşturulan bu model sınıfları belli olmayan ve sistem tarafından daha önceden karşılaşılmamış örnekler (test verisi) için sınıflandırma yapılmaktadır. Sınıflandırma problemlerine örnek olarak; bir e-postanın spam olup olmadığını belirlemek, tümörün kötü huylu olup olmadığını belirlemek ya da bir hayvanın görüntüsünü kedi, köpek veya başka bir hayvan olarak tanımlayabilmek verilebilir.

Tarihsel olarak, makine öğreniminin başlangıcı on yedinci yüzyıla, Pascal ve Leibniz tarafından insanın toplama ve çıkarma yeteneğini taklit edebilen makinelerin geliştirilmesine dayandığı söylenebilir. Modern tarihte, IBM'den Arthur Samuel "makine öğrenimi" terimini ortaya atmıştır ve bilgisayarların dama oynamayı öğrenmek üzere programlanabileceğini göstermiştir. Bunu, 1958'de nöral ağ mimarilerinden biri olarak Rosenblatt tarafından algılayıcının geliştirilmesi izlemektedir. Ancak, algılayıcıyla ilgili ilk yoğun ilgi, Minsky'nin algılayıcının sınıflandırma yeteneğinin doğrusal olarak ayrılabilir problemlerle sınırlı olduğu belirtmesiyle azaldı. 1975 yılında Werbos tarafından çok katmanlı algılayıcının (MLP) geliştirilmesiyle bir atılım sağlandı. Bunu, 1986 yılında Quinlan tarafından karar ağaçlarının geliştirilmesi ve Cortes ve Vapnik tarafından destek vektör makinelerinin geliştirilmesi izledi. Daha sonra Adaboost ve rastgele ormanlar dahil olmak üzere birden çok algoritmayı birleştiren topluluk makine öğrenimi algoritmaları önerilmiştir. Daha yakın zamanlarda, derin öğrenme kavramı altında dağıtılmış çok katmanlı öğrenme algoritmaları ortaya çıkmıştır (El Naqa ve Murphy, 2015: 5-6).

Makine öğrenmesi ile oluşturulan bir modelin kararlarını daha güvenilir hale getirmeye yönelik geliştirilen yaklaşım, birkaç farklı modelin çıktısını birleştirmektir. Farklı modelleri bir arada kullanmak nasıl iyi bir fikir olabilir? En iyi performansı gösteren bireysel modeli seçmekten daha iyisini nasıl yapabilirsiniz? Elbette tüm bunlar, Occam'ın usturası diye bilinen basitliği savunan ilkesine ters mi düşüyor? Epikuros'un, alternatif açıklamalarla karşı karşıya kalındığında, hepsinin saklanması gerektiği görüşünü de hatırlayalım. Akıllı (wise) insanlar kritik kararlar verirken, genellikle kendi kararlarına veya tek bir güvenilir danışmanın kararlarına güvenmek yerine, birkaç uzmanın görüşlerini dikkate almaktadırlar. Sadece bir uzmanın görüşünü körü körüne takip etmesi yanlış olduğundan farklı uzman görüşleri birleştirilebilir (Witten ve diğerleri, 2005: 183). Bu bakış açısıyla, bireysel model yerine

birden fazla modeli birleştiren topluluk öğrenmesi yöntemleri daha iyi performansa sahip midir? Her bir modeli bir uzman olduğu düşünüldüğünde, bireysel modeller karar için tek bir uzmandan görüş alındığı bir durum olarak düşünülebiliriz. Topluluk öğrenmesi algoritmaları genel olarak torbalama (bagging), artırma (boosting) ve yığın (stacking) algoritmalarıdır. Torbalama algoritmasında ise; tek bir uzman yerine, her uzmanın oy kullandığı bir görüşme panelinin olduğunu hayal edelim. Torbalama algoritmasında, demokratik bir oylama süreci aracılığıyla nihai karar için tüm uzmanlardan gelen girdilerin birleştirilmesini içerir. Artırma algoritmasında ise; her uzman önceki uzmandan gelen geri bildirimlere dayalı olarak değerlendirme kriterlerini değiştirdiği alternatif bir yaklaşım olarak düşünülebilir. Bu, daha dinamik bir değerlendirme süreci uygulayarak görüşme sürecinin verimliliğini artırmaktadır (Witten ve diğerleri, 2005: 351-353). Yığın algoritmasında ise nihai karar verilirken oylama prosedürünün yerine meta öğrenici kullanılmaktadır. Yapılan çalışmada, yığın algoritmasında kapsamlı bir model denemesi yapılarak performansı incelenmiştir.

Tezin amacı, sınıflandırma probleminin çözümünde kullanılan makine öğrenmesi algoritmalarının performansını iyileştirmektir. Algoritmaların performansı etkileyen en önemli unsur elimizdeki veri setindeki özniteliklerin yeterli sayıda ve gerekli olmasıdır. Bunun için sınıflandırma işlemi gerçekleştirilmeden önce öznitelik seçimi yapılmıştır. Öznitelik seçimi olarak da filtreleme ve sarmal yöntemleri birleştiren hibrit bir yaklaşım kullanılmıştır. Bu hibrit yaklaşımla, sadece filtreleme yönteminin kullanıldığı durumdaki sınıflandırma doğruluğunu iyileştirmekle kalmıyor, aynı zamanda sadece sarmal yöntemler kullanıldığı durumdaki işlem süresi de azaltılabilmektedir. Algoritmanın performansını geliştirmek için diğer bir denemiş yaklaşım ise, yığın algoritmasında temel ve meta model olarak farklı algoritmalar denenerek daha iyi performansa sahip model bileşimi bulunmaya çalışılmıştır. Yığın algoritmasında kullanılan temel ve meta modellere göre algoritmanın performansı değişmektedir. Temel ve meta model olarak en iyi performansı verebilecek algoritmalara ilişkin genel geçer bir kabul bulunmamaktadır. Farklı modeller denenerek en iyi performansa sahip model bulunmaya çalışılmaktadır.

Yapılan çalışmada, "GlaucomaM" veri seti ve veri bilimi projelerinde sıklıkla başvurulan veri kaynaklarından biri olan UCI Machine Learning Repository veri tabanındaki popüler veri setlerinden "Breast Cancer Wisconsin (Diagnostic)" adlı veri seti kullanılmıştır. Analizler R programlama dili kullanılarak yapılmıştır. Öznitelik seçimi için filtreleme adımında Ki-Kare, Bilgi Kazancı, Kazanç Oranı, Simetrik Belirsizlik Katsayısı, Korelasyon Tabanlı Öznitelik Seçimi ve RELIEF yöntemleri

kullanılırken; sarmal yöntem olarak Özyinelemeli Öznitelik Eliminasyonu, Genetik Algoritma, Tavlama Benzetimi ve BORUTA kullanılmıştır. Sınıflandırma işlemi K En Yakın Komşu, Lojistik Regresyon, Naive Bayes, Destek Vektör Makinaları, Doğrusal Diskriminant Analizi, Karar Ağaçları, Rasgele Orman, Torbalama, Artırma, Rotasyon Ormanı ve Yığın algoritmaları kullanılarak yapılmış ve model başarımları ölçütleri kullanılarak performansları kıyaslanmıştır.

Birinci bölümde makine öğrenmesi tanımı, öğrenme stratejileri, öznitelik seçim yöntemleri, makine öğrenmesi algoritmaları ve sınıflandırıcı performans değerlendirme ölçütleri açıklanmıştır. İkinci bölümde makine öğrenmesi alanında yazılmış tez ve makalelerin literatür incelemesi ve kullanılan veri setlerinin literatür incelemesi açıklanmıştır. Son olarak üçüncü bölümde kullanılan veri seti, algoritmaların performansını arttırmak için denenmiş yaklaşımlar, temel algoritma, yığın algoritmaları ve ardışık öznitelik seçimi sonrası tüm algoritmaların sonuçlarına değinilmiştir.

## BİRİNCİ BÖLÜM

### MAKİNE ÖĞRENMESİ

#### 1.1. MAKİNE ÖĞRENMESİ KAVRAMI

Makine öğrenmesini anlayabilmek için öncelikle öğrenme kavramının tanımlanması gerekmektedir. Öğrenme kavramı farklı şekillerde tanımlanmakla birlikte genellikle Simon tarafından önerilen tanım etrafında değişiklikler yapılmaktadır. Simon öğrenmeyi, "zaman içinde yeni bilgilerin keşfedilmesi yoluyla davranışların iyileştirilmesi süreci" olarak tanımlamaktadır. Makine öğrenmesi ise bu öğrenme işinin bilgisayarlar tarafından gerçekleştirilmesinin sağlanmasıdır. Makine öğrenmesi, sistemin geçmişteki deneyimlerinden elde edilen öğrenmelerini kullanarak bir model oluşturması ve gelecekte karşılaşılabilecek durumlar karşısında bir tahminde bulunulmasını sağlayan bir yapay zekâ alanıdır. Bilgisayarın eldeki veriler üzerinden geliştirilen bir model yardımıyla oluşturdukları öğrenmeleri, ileride karşılaştıkları yeni veriler üzerinde kullanarak kararlar verebilmesini ve ilgili problemlere çözümler üretmesi olarak tanımlanabilir. Başka bir deyişle makine öğrenmesi " bilgisayarın bir olay ile ilgili bilgileri ve tecrübeleri öğrenerek gelecekte oluşacak benzer olaylar hakkında kararlar verebilmesi ve problemlere çözümler üretebilmesidir" denilebilir (Bilgin, 2018: 13; Öztemel, 2003: 21).

Bilgisayar oyunları ve yapay zekâ alanında öncü kişilerden sayılan Arthur Lee Samuel 1959 yılında makine öğrenmesini, "bilgisayarların açıkça programlanmadan öğrenme yeteneği veren" bir çalışma alanı olarak tanımlamıştır. Samuel, IBM'deki çalışmalarında ilk kendi kendine öğrenen bilgisayar programlarından birini oluşturmasıyla tanınır. Bilgisayarın bir şeyler öğrenmesini sağlamanın bir yolu olarak oyunlara odaklanmıştır. Samuel dama oyununu tercih etmiştir. Dama oyunu basit ama programın öğrenebileceği bir strateji gerektirdiğinden tercih etmiştir (Bell, 2020: 2; Bilgin, 2018: 13).

Mitchell (1997) makine öğrenmesini bir bilgisayar yazılımı, deneyim E'yi kullanarak istenilen bir görev T'yi gerçekleştirmek için performans P ölçüsü ile eğitilmesidir. Burada performans P ölçütü ile oluşturulan görev T, deneyim E'yi geliştirmektir (Mitchell, 1997: 6). Tom M. Mitchell, Carnegie Mellon Üniversitesi'nde makine öğrenimi departmanının eski başkanıdır. Makine Öğrenimi (McGraw-Hill, 1997) kitabının yazarı olarak, makine öğrenimi tanımı sıklıkla alıntılanmaktadır.

Sözlükte "öğrenme" kelimesini şöyle tanımlıyor:

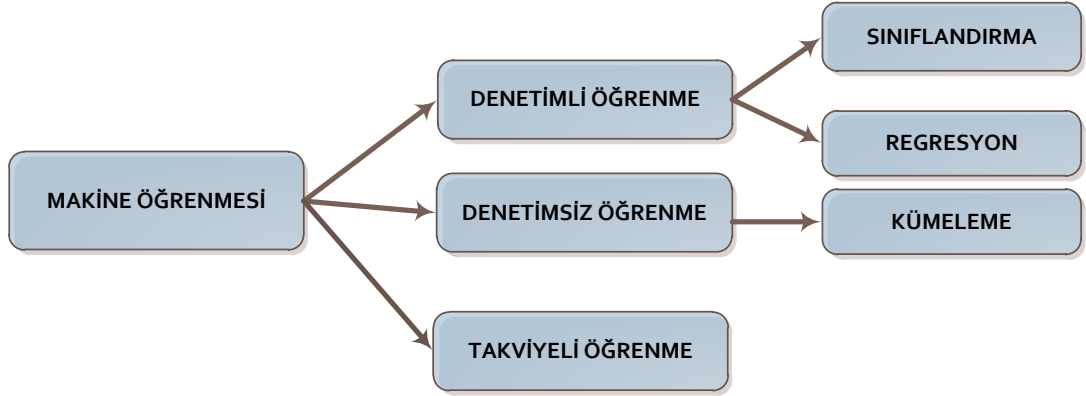
- Çalışma, deneyim veya öğretilme yoluyla bir şey hakkında bilgi edinmek
- Bilgi veya gözlem yoluyla farkında olmak
- Hafızaya almak için
- Hakkında bilgi sahibi olmak veya tespit etmek
- Talimat almak için

Bu sözlük anlamlarının bilgisayarlar söz konusu olduğunda bazı eksiklikleri bulunmaktadır. İlk ikisi için, öğrenmenin gerçekleşip gerçekleşmediğini test etmek neredeyse imkansızdır. Bir makinenin bir şey hakkında bilgi sahibi olup olmadığını nasıl anlarsınız? Muhtemelen ona sadece soru soramazsınız; sorsanız bile öğrenme yeteneğini değil, soruları yanıtlama yeteneğini test etmiş olursunuz. Bir şeyin farkında olup olmadığını nasıl anlarsınız? Bilgisayarların farkında mı yoksa bilinçli mi olabileceği konusu tartışmalı bir konudur. Son üç anlama gelince, insani terimlerle neyi ifade ettiklerini görebilsek de yalnızca hafızaya alma ve talimat alma, makine öğrenimi ile kastettiğimiz şeyin çok gerisinde kalıyor gibi görünüyor. Bunlar çok pasiftir ve bilgisayarların bu görevleri önemsiz bulduğunu biliyoruz. Bunun yerine, yeni durumlarda performanstaki iyileşmelerle ya da en azından performans potansiyeliyle ilgileniyoruz. Yeni bilgiyi yeni durumlara uygulayamadan ezberle öğrenme yoluyla bir şeyi hafızaya alabilir ya da bir şey hakkında bilgi sahibi olabilirsiniz. Başka bir deyişle, hiçbir fayda sağlamadan talimat alabilirsiniz (Witten ve diğerleri, 2005: 7).

## 1.2. ÖĞRENME STRATEJİLERİ

Örneklerden öğrenen sistemlerde değişik öğrenme stratejileri kullanılmaktadır. Öğrenmeyi gerçekleştirecek olan sistem ve kullanılan öğrenme algoritması bu stratejilere bağlı olarak değişmektedir (Öztemel, 2003: 25). Genel olarak öğrenme stratejileri; denetimli, denetimsiz ve takviyeli öğrenme olmak üzere üç grupta incelenmektedir.

**Şekil 1: Öğrenme Stratejileri**



Kaynak: Yazar tarafından oluşturulmuştur.

Genel olarak, denetimli istatistiksel öğrenme, bir veya daha fazla girdiye dayalı bir çıktıyı tahmin etmek veya tahmin etmek için istatistiksel bir model oluşturmayı içerir. Bu nitelikteki problemler; işletme, tıp, astrofizik ve kamu politikası gibi çeşitli alanlarda ortaya çıkabilmektedir. Denetimsiz istatistiksel öğrenmede girdiler vardır ancak denetleyici çıktılar yoktur; yine de bu tür verilerden ilişkileri ve yapıyı öğrenebilmekteyiz (James ve diğerleri, 2013: 1).

### 1.2.1. Denetimli Öğrenme

Denetimli öğrenmenin (supervised learning) amacı, bir dizi girdi değişkenine dayalı olarak çıktı değerini tahmin etmektir (Friedman ve diğerleri, 2001: 9). Bu stratejide öğrenilecek olayla ilgili örnekler girdi/çıkıtı seti olarak öğrenme sistemine verilerek eğitilmektedir. Yani hem her örnek için girdiler hem de bu girdiler için oluşturulması gereken çıktılar sisteme gösterilir. Sistemin görevi, girdileri belirlediği çıktılara eşlemektir. Bu şekilde girdiler ve çıktılar arasındaki ilişkiler öğrenilmektedir (Nilsson, 1996: 6). Model öğrenmeyi gerçekleştirerek, daha önce hiç görülmemiş veriler (test verisi) verildiğinde doğru etiketleme yapabilmesi sağlanmaktadır.

Denetimli öğrenme regresyon ve sınıflandırma problemleri olmak üzere iki alt grupta ele alınmaktadır. Regresyon, nicel çıktı değişkenleri tahmin etmek için kullanılırken; sınıflandırma, nitel çıktı değişkenini tahmin etmek için kullanılmaktadır (Chao, 2011: 9).

### 1.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenmenin (unsupervised learning) bir çıktı değeri yoktur ve amaç, bir dizi girdi değişkeni arasındaki ilişkileri ve örüntüleri (patterns) tanımlamaktır (Friedman ve diğerleri, 2001: 2). Bu strateji türünde sisteme sadece girdi değerleri gösterilir ve öğrenme işlemi sadece girdiler üzerinde gerçekleştirmeye çalışır. Sistemin örneklerdeki parametreler arasındaki ilişkileri öğrenerek ilişki ağını ortaya çıkarması beklenmektedir (Öztemel, 2003: 25).

Denetimsiz öğrenme, genellikle kümeleme, öznitelikler arasındaki ilişkileri bulma, olasılık yoğunluk tahmini ve boyut indirgeme gibi amaçlarla kullanılmaktadır (Chao, 2011: 9).

### 1.2.3. Takviyeli Öğrenme

Takviyeli (pekiştirmeli-reinforcement) öğrenme, istenen davranışları ödüllendirmeye ve/veya istenmeyenleri cezalandırmaya dayanan bir makine öğrenmesi eğitim yöntemidir. Genel olarak, bir pekiştirmeli öğrenme ajanı çevresini algılayabilir, yorumlayabilir, harekete geçebilir ve deneme yanılma yoluyla öğrenebilir. Henüz pratikteki kullanımları sınırlı olsa da gelişmeye açık bir yöntemdir.

Bu stratejide öğrenen sisteme, her girdi seti için olması gereken (üretilmesi gereken) çıktı setini sisteme göstermek yerine sistemin kedisine gösterilen girdilere karşılık çıktısını üretmesi beklenir ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Sistem bu sinyali dikkate alarak öğrenme sürecini devam ettirir (Öztemel, 2003: 25). Bu stratejiye örnek olarak robot algısı ve hareketi, otomatik satranç oyuncusu, otomatik araç sürüşü vb. verilebilir (Chao, 2011: 9).

## 1.3. SINIFLANDIRMA

Sınıflandırma (classification), elimizdeki verilerin sınıflarının belli olduğu durumda, sınıfı belli olmayan yeni verilerin sınıflarının bulunması işlemidir. Amaç, elimizde sınıfları belli olan verileri (eğitim verisi) kullanarak oluşturulan modelin, sınıfları belli olmayan ve sistem tarafından daha önceden karşılaşılmamış örnekler (test verisi) üzerinde yüksek doğrulukta sınıflandırma yapmasını sağlamaktır (Silahtaroglu, 2016: 67). Sınıflandırma hem girdi değişkeni (bağımsız değişken) hem de çıktı (bağımlı değişken) için değerleri içeren bir dizi gözlemle başlar. Bağımlı

değişken, bağımsız değişkenlere dayalı olarak tahmin etmeyi umduğumuz özelliktir (öznitelik). Bağımsız değişken, ilgilenilen sonuç değişkeni ile potansiyel olarak ilişkili olan özelliklerdir (öznitelikler). Veri kümesi daha sonra farklı yaklaşımlarla eğitim ve test seti olmak üzere bölünür. Eğitim setindeki veriler kullanılarak tahmine dayalı model geliştirilir ve test setindeki veriler kullanılarak doğruluk açısından test edilir. Etkili bir model oluşturduktan sonra, bunu yalnızca açıklayıcı değişkenlerin (girdi değişkenleri) bulunduğu durumlarda çıktı değerini tahmin etmek için kullanılabilir (Lewis ve Street, 2000: 2-3). Oluşturulan model, sınıflandırma kuralları (yani, IF-THEN kuralları), karar ağaçları, matematiksel formüller veya sinir ağları gibi çeşitli şekillerde temsil edilebilmektedir. Resim, örüntü tanıma, hastalık tanısı, dolandırıcılık tespiti, kalite kontrol, pazarlama ve finans gibi birçok alanda yaygın olarak kullanılmaktadır.

Sınıflandırma işleminin iki aşaması bulunmaktadır (Aggarwal, 2015: 286-287):

1. Eğitim aşaması: Bu aşamada, eğitim verileri (sınıfları etiketlenmiş verileri) kullanılarak bir model oluşturulur. Bu eğitim veri setindeki etiketli verilerin bir matematiksel modeli olarak anlaşılabilir.

2. Test aşaması: Bu aşamada, bir veya daha fazla görülmeyen test örneğinin (etiketsiz veri) sınıf etiketi (çıktı değeri) eğitim modeli kullanılarak belirlenmektedir.

#### 1.4. ÖZİNİTELİK SEÇİM YÖNTEMLERİ

Veri setlerindeki öznitelikler (değişken/nitelik/özellik) sınıflama performansını etkileyen en önemli unsurlardan biridir. Bir veri setinde toplanan özniteliklerin sayısı genellikle nispeten büyüktür (yani büyük boyutun laneti) ve bu özelliklerin tümü bilgilendirici değildir veya yüksek ayırt etme gücü sağlamamaktadır (Tsai ve diğerleri, 2013: 241). Birçok gerçek hayat probleminin çözümü için kullanılan verilerde gereksiz, ilgisiz, gürültülü, yanıltıcı vs. değişkenler yer almakta ve problemlerin çözümü için kurulacak modeller hakkında genellikle önsel bilgi bulunmamaktadır. Bazı makine öğrenme algoritmaları, gereksiz, ilgisiz, gürültülü, yanıltıcı gibi özniteliklere karşı duyarlıdır ve bu tür algoritmalarda performans düşüşleri ortaya çıkabilmektedir. Bu nedenle özniteliklerin yeterli sayıda ve doğru olarak belirlenmesi gerekmektedir (Budak, 2018: 21-22; Khalid ve diğerleri, 2014: 372-373).

Öznitelik seçimi (feature selection), bir veri kümesindeki modeli oluşturmak için verilen orijinal özellik kümesinden alakasız ve gereksiz özellikleri ortadan kaldırarak orijinal veri setini temsil edebilecek en iyi altkümenin seçimi olarak

tanımlanmaktadır. En iyi alt küme, öğrenme doğruluğuna en çok katkıda bulunan en az sayıda boyuta sahip olandır. Özellik seçimi (diğer adıyla nitelik seçimi veya değışken seçimi), kullanılan algoritmaya göre özellikleri deęerlendirerek veri setindeki n adet özellik arasından en iyi k adet özellięi seçme işlemidir (Budak, 2018: 21-22; Khalid ve dięerleri, 2014: 373). Gerçekleştiren bu işlem ile gereksiz ya da çok fazla ayırt edici nitelik barındırmayan özelliklerin elenmesi sağlanmaktadır. Günlük hayatta karşılaştığımız gerçek problemlerin çoęu yanıltıcı veya gürültü adı verilen bileşenlere sahip olduęu düşünöldüğünde yapılan bu işlem aslında bir gerekliliktir (Bilgin, 2018: 24).

Öznitelik seçimi analiz sürecinde uygulamacıya birçok avantaj sağlamaktadır (Khalid ve dięerleri, 2014: 373):

- ✓ Öznitelik kümesinin boyutunu düşürdüğümüzde, öğrenme süresi kısılır böylece algoritma hızı artar,
- ✓ Gereksiz, ilgisiz veya gürültülü veriyi ortadan kaldırarak veri kalitesini iyileştirir,
- ✓ Veri kümesini daha basit bir şekilde tanımlanabilir, görselleştirilebilir ve anlaşılabilir hale getirir, yani modelin karmaşıklığını en basit düzeye indirgemiş olur,
- ✓ Veri depolamak için gerekli olan hafıza miktarını azaltır,
- ✓ Elde edilen modelin performansını artırır,
- ✓ Ezberleme riskini de azaltır. Böylece; aşırı uyum (overfitting) problemi aşılır.
- ✓ Sonuçların anlaşılabilirliğini iyileştirir.

Öznitelik seçiminin dezavantajı ise aşağıdaki gibi belirtilebilir (Hall, 1999: 3):

- ✗ Hesaplama süresi açısından, en iyi alt kümeyi seçmek, ek hesaplama maliyeti gerektirir.

#### **1.4.1. Filtreleme Yöntemleri**

Filtreleme (filter) yöntemleri, özniteliklerin modelin açıklama/doęrulama gücüne etkisinin tek tek analiz edilmesine dayanan en eski öznitelik seçim yöntemi olarak bilinmektedir. Bu yöntemlerde öznitelik seçimi, herhangi bir sınıflandırma algoritması kullanılmadan, uzaklık, bilgi, bağımlılık ve tutarlılık ölçümleri gibi istatistiksel kriterlere dayanan fonksiyonlar kullanılarak yapılmaktadır (Saeys ve dięerleri, 2007: 2507-2508). Öznitelik seçme ve sınıflandırma işlemleri birbirinden ayrı ve bağımsız olarak gerçekleştirilmektedir. Benzer mantıkla çalışan bu yöntemlerde, her bir öznitelik için istatistiksel fonksiyonlar kullanılarak bir deęer hesaplanmakta ve

hesaplanan bu değerler arasından en yüksek değere sahip olan öznitelikler seçilmektedir. Seçilen öznitelikler sınıflandırma algoritmaları için girdi verileri olarak sunulur ve bu öznitelikler kullanılarak sınıflandırma işleminin performansı değerlendirilir (Subanya ve Rajalaxmi, 2014: 2). Filtreleme yöntemlerine örnek olarak Fisher Skor, Ki-kare, Bilgi Kazancı, Kazanç Oranı, F-Skor, Simetrik Belirsizlik Katsayısı, Korelasyon Tabanlı Öznitelik Seçimi, RELIEFF verilebilir.

**Ki-Kare Yöntemi:** Kategorik değişkenler arasında ilişkinin araştırılmasında kullanılan ki-kare test istatistiği öznitelik seçimi için de sık kullanılmaktadır. Bu yöntemde, girdiler ile çıktı arasında ilişki olup olmadığını test edilmektedir. Yapılan test sonucunda, çıktı ile ilişkisi olmayan girdiler belirlenerek veri setinden çıkartılır. Bunun için çıktı ile girdiler arasındaki ki-kare değeri aşağıdaki formül kullanılarak hesaplanır. Ki-kare değeri ilgili girdinin önem derecesini göstermektedir. En büyük ki-kare değerine sahip girdi, çıktı değişkeni üzerinde en fazla ayırt ediciliğe sahip özniteliktir. (Budak, 2018: 24; Altunkaynak, 2019: 24-25)

$$\chi^2 = \sum_{i=1}^n \frac{(o_i - e_i)^2}{e_i}$$

$n =$  veri setindeki öznitelik sayısı

$o_i = i$ 'inci öznitelik için gözlenen frekans

$e_i = i$ 'inci öznitelik için beklenen frekans

**Bilgi Kazancı (Information Gain-IG):** Karar ağaçlarında ağaçtaki dallanmanın hangi kritere göre yapılacağı veya hangi öznitelik değerlerine göre ağaç yapısının oluşturulacağını belirleyen kriterlerdendir. Aynı zamanda filtreleme öznitelik seçim yöntemi olarak da yaygın kullanılmaktadır. Bu yöntem, sınıflar hakkında en fazla bilgiye sahip özellikleri seçmektir. Entropi tabanlı bir öznitelik seçme algoritması olup her öznitelik için IG katsayısı aşağıdaki formül kullanılarak hesaplanıp, en yüksek IG katsayısına sahip öznitelik kümesi seçilmektedir (Gümüüşçü ve diğerleri, 2016: 3).

$$H(Y) = - \sum_{y \in Y} p(y) \log_2(p(y))$$

$$H(Y/X) = - \sum_{x \in X} p(x) \sum_{y \in Y} p(y/x) \log_2(p(y/x))$$

$$\text{Bilgi Kazancı} = H(Y) - H(Y/X)$$

Kazanç Oranı (Gain Ratio-GR): Bilgi kazancı ölçütü çok çeşitli değerlere sahip öznitelikleri seçme eğiliminde olduğundan bu probleme çözüm olarak önerilen entropi tabanlı bir yöntemdir. Kazanç oranı bu probleme çözüm olarak her bir öznitelik için bölünme bilgisini hesaplar ve elde edilen bilgi kazancını bölünme bilgisine bölerek kazanç oranını hesaplanmaktadır. Kazanç oranı aşağıdaki gibi hesaplanmaktadır (Budak, 2018: 25):

$$\text{Kazanç Oranı} = \frac{\text{Bilgi Kazancı}}{H(X)}$$

Simetrik Belirsizlik Katsayısı (Symmetrical Uncertainty-SU): Bilgi kazancının olumsuz yanını giderebilmek için bilgi kazancını Y ve X'in entropi değerlerinin toplamına bölen entropi tabanlı bir yöntemdir. SU katsayısı aşağıdaki formül kullanılarak hesaplanıp, en yüksek SU katsayısına sahip öznitelik kümesi seçilmektedir. SU değeri 0 olan öznitelik, öğrenme modelini etkileyemeyeceği için göz ardı edilir (Budak, 2018: 25)

$$SU = 2 \frac{\text{Bilgi Kazancı}}{H(Y) + H(X)}$$

Korelasyon Tabanlı Öznitelik Seçimi (Correlation based Feature Selection-CFS): CFS her özneliğin sınıf etiketini tahmin etmekteki başarısının yanı sıra aralarındaki iç korelasyon değerlerini de dikkate almaktadır. Bu yaklaşım, sınıf etiketleri ile yüksek birbirleri ile düşük korelasyona sahip öznitelik alt kümesini seçme prensibine dayanmaktadır. Sınıf etiketiyle düşük korelasyona sahip olan alakasız öznitelikleri göz ardı etmektedir. Aynı zamanda diğer özniteliklerle yüksek oranda ilişkili olan öznitelikler sonuç öznitelik alt kümesinde dikkate alınmaz (Chaudhary ve diğerleri, 2016: 216). Aşağıdaki formül ile hesaplanmaktadır:

$$M_s = \frac{k\bar{r}_{ci}}{\sqrt{k + k(k-1)\bar{r}_{ii}}}$$

$s = k$  adet özneliğe sahip öznitelik alt küme sayısı

$\bar{r}_{ci}$  = sınıf ile öznitelikler arasındaki korelasyon

$\bar{r}_{ii}$  = özniteliklerin birbirleri arasındaki korelasyon

RELİEF: Bu algoritmanın ana fikri k-en yakın komşu algoritmasının kurallarına benzemekte olup, veri setindeki bir örnek ele alınarak ilgili örneğin, kendi sınıflarındaki

diğer örneklerle yakınlığını ve farklı sınıflarla olan uzaklığına bağlı olarak model oluşturularak öznitelik seçme işlemi gerçekleştirilmektedir. Yöntemin ilk adımında tüm öznitelikler 0 ağırlık değerine sahiptir. Daha sonra her bir adımda veri setinden rastgele örnek seçilerek, bu örnek ile aynı sınıfa sahip en yakın örnekteki ilgili öznitelik değeri ve farklı sınıfa sahip en yakın örnekteki ilgili öznitelik bulunur. Sonrasında ilgili öznitelige ait ağırlıklar hesaplanır. Öznitelikler ağırlıklarına göre sıralanır ve en iyi n adet öznitelik seçilir (Kaynar ve diğerleri, 2018: 181).

Belirli bir özniteligin ağırlığı (önem derecesi)  $W$  aşağıdaki gibi hesaplanır (Peker ve diğerleri, 2015: 3).

$$(W = W - \text{diff}(x_{ij}, \text{near}_{hit_{ij}})^2 + \text{diff}(x_{ij}, \text{near}_{miss_{ij}})^2) / m$$

$m$ : örneklem büyüklüğü

$\text{diff}(x_{ij}, \text{near}_{hit_{ij}})$  : rastgele seçilen  $j$  mesafesi içindeki öznitelik değerleri ile aynı sınıftaki en yakın eğitim verisi içindeki özniteligin ( $\text{near}_{hit_{ij}}$ ) değeri arasındaki fark

$(x_{ij}, \text{near}_{miss_{ij}})$ : farklı sınıftan en yakın eğitim örneğinin değeri olarak tanımlanır

#### 1.4.2. Sarmal Yöntemler

Sarmal (wrapper) yöntemlerde, özellik seçimi için doğru sınıflandırma oranına bağlı olarak verimliliğin ölçüldüğü çeşitli öğrenme algoritmaları kullanılarak en iyi tahmin performansını gösteren özellikler seçilmektedir (Budak, 2018: 25). Bir başka deyişle, sarmal yöntemde makine öğrenme algoritmaları kullanılır ve nitelik seçme ölçüsü sınıflandırıcının doğruluk oranıdır. Her bir iterasyonda belirli nitelik altkütmesi için sınıflandırma sonucu elde edilir (Saeys ve diğerleri, 2007: 2509). Sarmal yöntemlere örnek olarak: Ardışık ileri yönde seçim (Sequential Forward Selection- SFS), ardışık geri yönde seçim (Sequential Backward Selection - SBS), ardışık ileri yönde kayan seçim (Sequential Forward Floating Selection- SFFS), ardışık geri yönde kayan seçim (Sequential Backward Floating Selection- SBFS), özimizelemeli özellik eliminasyonu (Recursive Feature Elimination- RFE), genetik algoritma (Genetic Algorithm- GA), Tavlama Benzetimi (Simulated Annealing – SA) ve BORUTA verilebilir.

Sarmal yöntemlerde, optimal öznitelik alt kümesini bulmak için bir arama yöntemi kullanılmaktadır. Bu yöntem, arama stratejisine göre genel olarak iki

kategoriye ayrılır: Sıralı arama ve rastgele arama. Sıralı arama yöntemleri, özellikleri sırayla ekler veya kaldırır, ancak yerel bir optimuma sıkışıp kalma eğilimindedirler. Örnek olarak sıralı geriye doğru seçim, sıralı ileri seçim ve kayan arama yöntemleri verilebilir. Öte yandan rastgele arama yöntemleri, yerel optimum çözümlerden kaçmak için arama prosedürlerine rastgeleliği dahil etmeye çalışır. Bu yöntemlerin örnekleri arasında tavlama benzetimi ve genetik algoritma optimizasyonu yer almaktadır (Tabakhi ve diğerleri, 2014: 114).

Özyinelemeli Öznitelik Eliminasyonu (Recursive Feature Elimination-RFE): RFE yönteminde, en zayıf özneliği (veya öznelikleri) belirtilen öznelik sayısına ulaşılan kadar özyinelemeli olarak ortadan kaldırarak, orijinal veri kümesini temsil edebilecek en iyi alt kümenin belirlenmesini sağlar. Özyineleme gereklidir, çünkü bazı ölçütler için her bir özneliğin göreceli önemi, aşamalı eleme işlemi sırasında farklı bir öznelik alt kümesi üzerinden değerlendirildiğinde büyük ölçüde değişebilir (özellikle yüksek oranda ilişkili öznelikler için) (Granitto, 2006: 84).

Bu yöntemde, ilk olarak tüm öznelikler kullanılarak bir model eğitilip her öznelik için önem skoru hesaplanmaktadır. Daha sonra en az önem skoruna sahip öznelikler göz ardı edilerek model yeniden eğitilerek önem skorları hesaplanır. Bu işlem öznelik kümesinde başlangıçta belirlenen istenilen sayıda öznelik elde edilinceye kadar devam ettirilir. Öznelik seçimi sonunda istenilen öznelik sayısı ve öznelik önem skorlarının hesaplanacağı modelin seçimi ayarlanması gereken parametrelerdir (Gündüz, 2019: 1168).

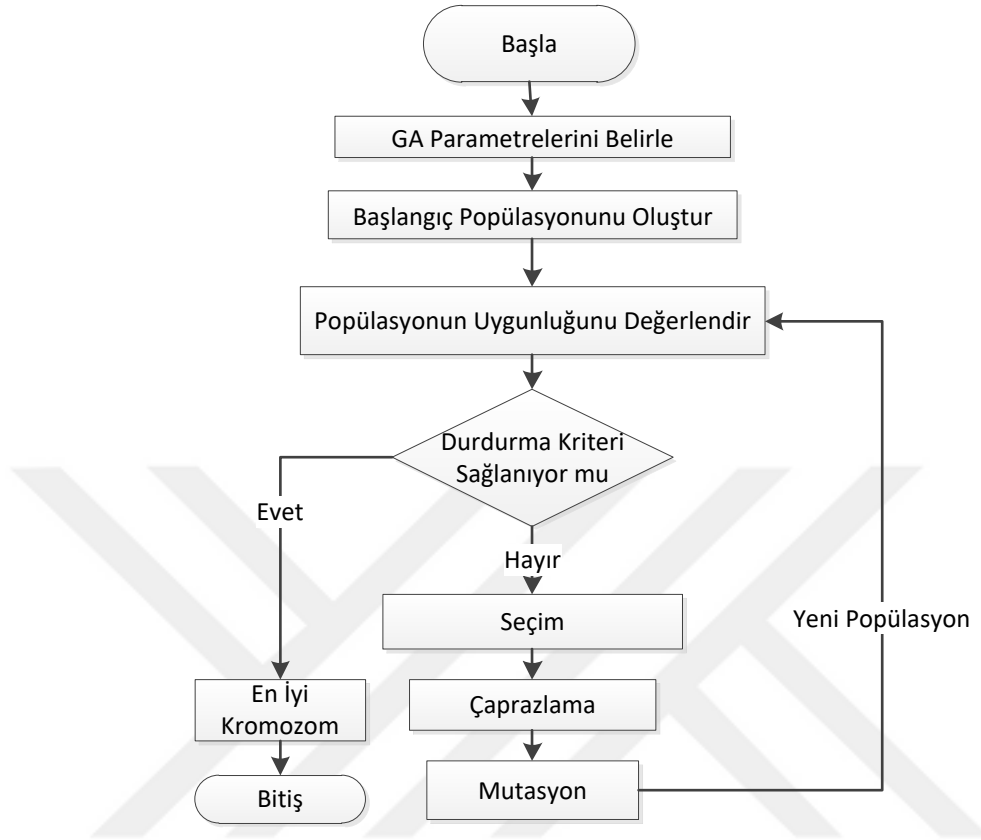
BORUTA: Boruta, rastgele orman algoritmasına dayalı bir öznelik seçim algoritmasıdır. Algoritma, ilk olarak orijinal veri kümesine tüm özneliklerin kopyaları (gölge öznelik de denilen) eklenerek veri seti genişletilir. Hedef (çıktı-yanıt) değişken ile korelasyonu kaldırmak bir başka deyişle rastgelelik elde etmek için eklenen öznelikler karıştırılır. Daha sonra, boyutu orijinal veri kümesinin iki katı olan yeni bir öznelik uzayını elde etmek için gölge öznelikler orijinal özneliklerle birleştirilir. Genişletilmiş veri setinde rastgele orman sınıflandırıcı eğitilir ve Z skorları hesaplanır. Gölge öznelikler arasından en yüksek Z skoru (MZSA) bulunur ve gerçek (orijinal) özneliklerin, gölge özneliklerin maksimum Z puanından daha yüksek bir Z puanına sahip olup olmadığı kontrol edilir. MZSA'dan önemli ölçüde daha düşük öneme sahip öznelikler 'önemsiz' olarak kabul edilir ve bilgi sisteminden kalıcı olarak kaldırılır. MZSA'dan önemli ölçüde daha yüksek öneme sahip öznelikler 'önemli' olarak kabul edilir. Ardından tüm gölge öznelikler veri setinden kaldırılır. Tüm öznelikler için

önem atanana veya algoritma önceden belirlenmiş rastgele orman çalıştırma sınırına ulaşana kadar işlemler tekrarlanır (Kursa ve Rudnicki, 2010: 3-4)

Genetik Algoritma (Genetic Algorithm-GA): GA Darwin'in evrim teorisinden ilham alınan, 1975 yılında Holland tarafından matematiksel olarak formüle edilen sezgisel arama algoritmasıdır. Her gen bir değişkeni temsil eder ve bir gen dizisine kromozom (dizi, birey) denir. Her bir kromozom, optimizasyon probleminde potansiyel bir çözümü ifade etmektedir (Pulat ve Kocakoç 2017: 22). Evrim genellikle rastgele oluşturulmuş bireylerden oluşan bir popülasyondan başlar ve nesiller boyunca gerçekleşir. İlk olarak, rastgele oluşturulmuş bir kromozom popülasyonu başlatılır. Kromozomların ne kadar iyi olduğu önceden belirlenmiş bir uygunluk fonksiyonuna göre değerlendirilir. Yeni popülasyon daha uygun bireylerin seçimi ile oluşur. Bu yeni popülasyondaki bireyler çaprazlama ve mutasyon ile değişime uğrar ve yeni çözümler oluşur. GA belirli bir nesil sayısı veya durdurma kriteri sağlanana kadar defalarca çalıştırılır. GA işleyişi sonucunda en iyi kromozom (dizi, birey) çözüm olarak alınır (Liu ve Fu, 2012: 8-9; Pulat ve Kocakoç 2019: 228). GA'nın akış diyagramı Şekil 2'de verilmiştir.

GA bir optimizasyon algoritması olarak öznitelik seçiminde sıklıkla kullanılmaktadır. İlk olarak kromozomların nasıl kodlanacağına karar vermek gerekmektedir. Bireyin nasıl kodlanacağı ilgilenilen probleme göre değişiklik göstermektedir. Öznitelik seçiminde bireyler, genellikle ikili (binary, bit string) olarak kodlanmaktadır. Bit değeri 0 ise ilgili öznitelik seçilmemiş, eğer 1 ise ilgili öznitelik seçilmiş olduğunu göstermektedir. Ardından popülasyondaki her bireyin uygunluğunu değerlendirmek için sınıflandırma algoritması eğitilmekte ve sınıflandırıcının performans kriterine göre başlangıç popülasyonundan bireyler seçilmektedir. Daha sonra seçilen bireylere çaprazlama ardından mutasyon operatörleri uygulanarak yeni çözümler oluşturulur. Oluşturulan yeni çözümlerin uygunluğu değerlendirilerek seçim, çaprazlama ve mutasyon işlemi önceden belirlenmiş bir durdurma kriteri sağlanana kadar işlemler tekrar edilir (Albayrak ve Albayrak, 2016: 2).

**Şekil 2:** Genetik Algoritma Akış Diyagramı



Kaynak: Pulat ve Kocakoç 2019: 228.

Fitre yöntemlerinin en büyük avantajı hesaplama zamanı açısından sarmal yöntemlere göre oldukça hızlı olmasıdır. En büyük dezavantajı ise seçim kriterinin doğrudan model etkinliğiyle ilgili olmamasıdır; bu yüzden sınıflandırma performansı zayıftır. Şöyle ki, bir sınıf ile yüksek korelasyona sahip özelliklerin seçimi bu özelliklerin bir araya gelerek oluşturduğu modelin anlamlı olmamasıyla sonuçlanabilir. Çünkü böyle bir durumda özellikler arasındaki etkileşimler dikkate alınmaz. Sarmal yöntemlerin en büyük avantajı doğrudan modeli iyileştiren özellikler kümesinin seçimine imkân sağlamasıdır. Ancak bu yöntemler çoğunlukla sezgisel algoritmalara dayalı oldukları için kullanılan algoritmanın girdi parametrelerinin ayarlanması (parameter tuning) önemlidir (Altunkaynak, 2017: 22; Saeys ve diğerleri, 2007: 2509).

## 1.5. KULLANILAN MAKİNE ÖĞRENMESİ ALGORİTMALARI

Tez çalışmasında kullanılan makine öğrenmesi algoritmalarına bu bölümde kısaca değinilmiştir. Literatürde bulunan algoritmaların neredeyse tamamı tez kapsamında kullanılmıştır.

### 1.5.1. K-En Yakın Komşu

En yakın komşu algoritması (K- nearest neighbour-KNN) örnek kümesindeki gözlemlerin her birinin, sınıfı belirlenmek istenen gözlem değerine olan uzaklıklarının hesaplanması ve en küçük uzaklığa sahip k sayıda gözlemin seçilmesi esasına dayanan parametrik olmayan bir yöntemdir (Özkan, 2016: 141). Uzaklık hesaplamasında aşağıda verilen Öklid uzaklık formülü kullanılabilir:

$$d(i, j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}$$

Yöntemin adımları şu şekildedir (Özkan, 2016: 142):

- 1.adım: Verilen bir noktaya en yakın komşu sayısı olan k parametresi belirlenir.
- 2.adım: Sınıfı belirlenmek istenen noktanın, eğitim setinde yer alan diğer tüm noktalar arasındaki uzaklıkları hesaplanır.
- 3.adım: Hesaplanan uzaklıklar sıralanır ve en küçük k tanesi seçilir.
- 4.adım: Seçilen k tane satırın ait olduğu sınıflar belirlenir ve en çok tekrar eden (çoğunluk oylaması) kategorisi bilinmeyen noktanın kategorisi olarak kabul edilir.

### 1.5.2. Naive Bayes

Naive bayes (NB) algoritması, verilen bir veri setindeki değerlerin frekansını ve kombinasyonlarını sayarak olasılıkların bir kümesini hesaplayan basit bir olasılıksal sınıflandırıcıdır. Algoritma bayes teoremini kullanılır ve tüm değişkenlerin sınıf değişkenininin değeri göz önüne alındığından bağımsız olduğunu varsaymaktadır (Patil ve Sherekar, 2013: 257).  $X = \{x_1, x_2, \dots, x_n\}$  öznitelik değerinden oluşan sınıf üyeliği bilinmeyen veri örneği olsun ve m tane sınıf olduğunu varsayalım.  $C_1, C_2, \dots, C_n$  sınıf değerleri ile ilgili olasılıklar aşağıda verilen denklemle hesaplanır (Özkan, 2016: 157-158):

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

Örneğe ait  $x_i$  değerlerinin birbirinden bağımsız olduğu varsayıldığından aşağıdaki denklem kullanılabilir:

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

Bilinmeyen örnek  $X$ 'i sınıflandırmak için yukarıda hesaplanan olasılık değerlerinden en büyük olanı seçilerek sınıfı bilinmeyen örneğin ait olduğu sınıf olarak belirlenmektedir, aşağıdaki denklemde gösterilmiştir.

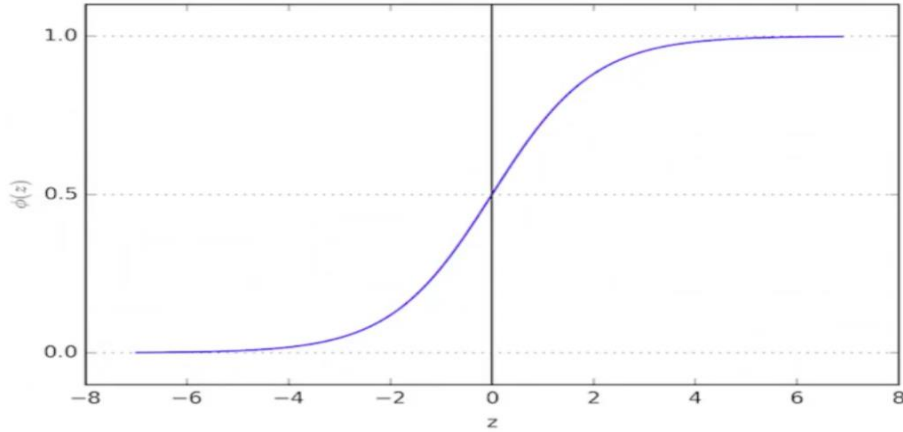
$$\arg \max_c \{P(C_i|X)\}$$

### 1.5.3. Lojistik Regresyon

Lojistik regresyonda (Logistic Regression-LR) temel amaç, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi açıklayabilen kabul edilebilir bir model kurabilmektir. LR, doğrusal regresyonun varsayımlarından hiçbirini gerektirmediği için büyük bir esneklik sağlamaktadır. Lojistik regresyonun, doğrusal regresyonla arasındaki en önemli farklılık ise; lojistik regresyonda bağımlı değişkenin kategorik olmasıdır. Bir diğer farklılık ise, doğrusal regresyon analizinde bağımlı değişkenin alacağı değer kestirilirken, lojistik regresyon analizinde ise bağımlı değişkenin alabileceği değerlerden birinin gerçekleşme olasılığı kestirilmektedir. Lojistik regresyonun temeli aşağıdaki denklemde verilen lojistik fonksiyona dayanmaktadır ve lojistik fonksiyona ait grafik Şekil 3'te verilmiştir (Balaban ve Kartal, 2015: 73-75).

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

**Şekil 3:** Lojistik Fonksiyon (Logistic Function)



Kaynak: Mlxtend, [https://rasbt.github.io/mlxtend/user\\_guide/classifier/LogisticRegression/](https://rasbt.github.io/mlxtend/user_guide/classifier/LogisticRegression/), 15.05.2023.

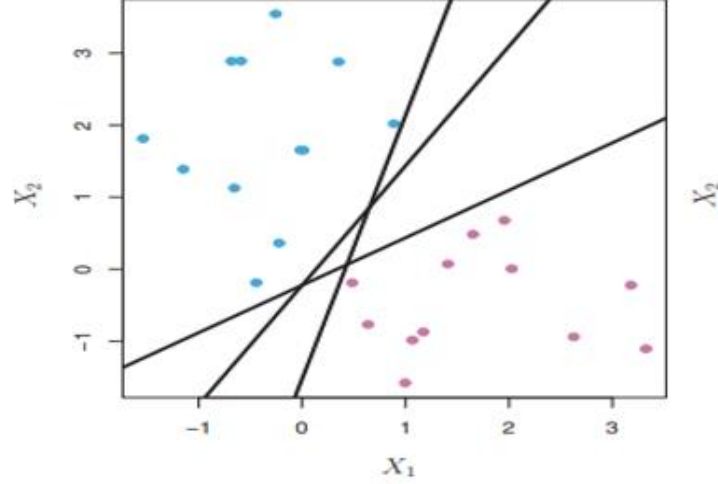
0 ile 1 arasında değer alabilen lojistik fonksiyon değerinin 0.5 ve üzerinde olması halinde 1'e yaklaştığı, aksi takdirde 0'a yaklaştığı öngörülmektedir. Böylece sınıf değerinin 1'e yakınsama olasılığı da hesaplanır. Olasılıkları hesaplanan sınıf değerlerini karşılaştırmak için üstünlük değeri (odds) hesaplanır (Balaban ve Kartal, 2015: 75). P, bir olayın olasılığı ise olayın üstünlük değerinin hesaplanması aşağıdaki formülle yapılmaktadır:

$$\text{Odds değeri} = \frac{p}{1 - p}$$

#### 1.5.4. Destek Vektör Makinesi

Destek Vektör Makinesi (Support Vector Machine-SVM) sınıflandırmasının ilk amacı, iki sınıfı birbirinden ayıracak doğruyu (hiper düzlem) belirlemektir. Bir başka deyişle, genelleme hatasını en aza indirmek ve böylece marjı en üst düzeye çıkarmak için iki sınıf arasında optimal bir ayırıcı hiper düzlem oluşturmaktır. Eğer herhangi iki sınıf, sonsuz sayıda hiper düzlem tarafından ayrılabilir ise, (Şekil 4'te gösterildiği gibi) SVM, genelleme hatasını (yani, görünmeyen test desenleri için hata) en aza indiren hiper düzlemi belirlemeye çalışmaktadır.

**Şekil 4:** Mavi ve Mor ile Gösterilen İki Sınıf için Pek Çok Olası Hiper Düzlem Arasından Üç Ayırıcı Hiper Düzlem



Kaynak: James ve diğerleri, 2013: 340.

Verinin doğrusal olarak ayrılmadığı durumlarda doğrusal sınıflandırıcı yerine doğrusal olmayan sınıflandırıcılar kullanılabilir. SVM' de doğrusal olarak verinin ayrıt edilemediği durumlarda (Şekil 5'te gösterildiği gibi) veriyi daha büyük boyutlu uzaylara taşıyarak çözümlmek için çekirdek (kernel) fonksiyonlardan yararlanılmaktadır. Çekirdek fonksiyonun seçimi çok önemlidir ve farklı çekirdek fonksiyonların seçilmesiyle farklı performanslar elde edilebilir (Deka, 2014: 374). Sıklıkla kullanılan çekirdek fonksiyonlar aşağıda verilmiştir (Hussain ve diğerleri, 2011 :147).

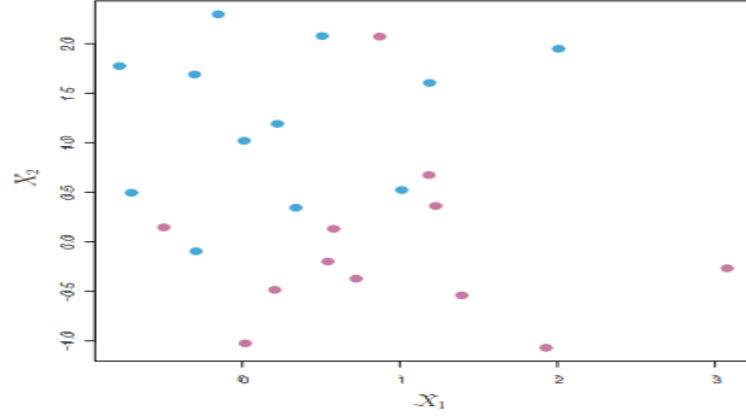
Doğrusal çekirdek fonksiyon:  $K(x_i, x_j) = (x_i^T x_j)$

d. derece polinom çekirdek fonksiyon:  $K(x_i, x_j) = (\gamma x_i^T x_j + 1)^d$

Radyal tabanlı çekirdek fonksiyon:  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$

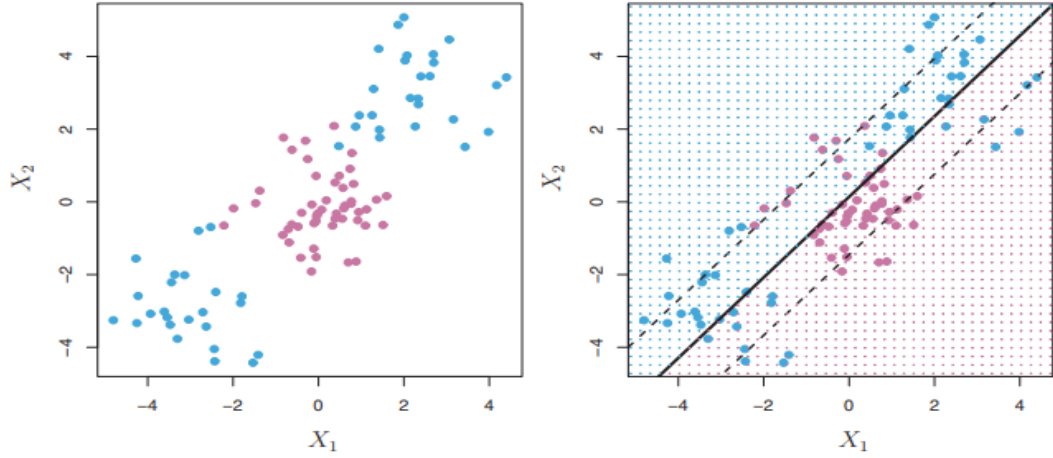
Sigmoid çekirdek fonksiyon:  $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

**Şekil 5:** Mavi ve Mor ile Gösterilen İki Sınıfına İlişkin Örnek



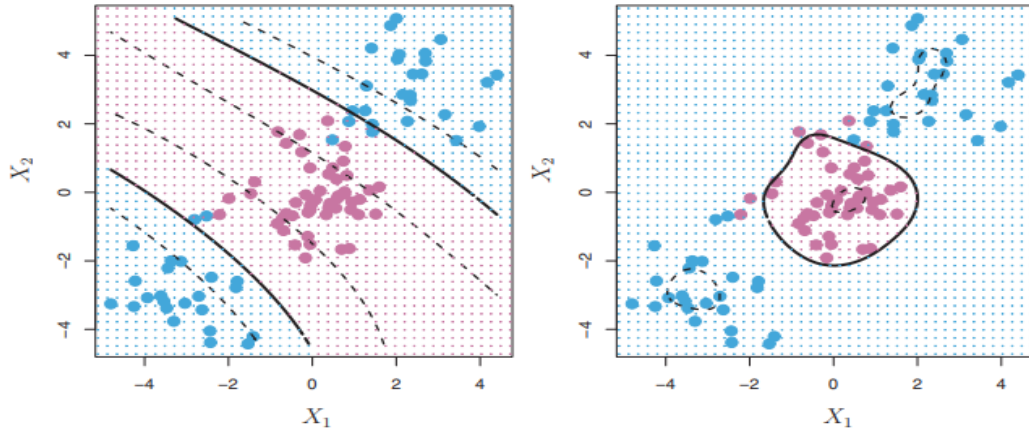
Kaynak: James ve diğerleri, 2013: 344.

**Şekil 6:** Mavi ve Mor ile Gösterilen İki Sınıfına İlişkin Örnek



Kaynak: James ve diğerleri, 2013: 349.

**Şekil 7:** Mavi ve Mor ile Gösterilen Sınıfına Ait Örneklerde Farklı çekirdek Fonksiyon Kullanımı



Kaynak: James ve diğerleri, 2013: 353.

Şekil 5'te iki değişkene ilişkin mavi ve mor ile gösterilen iki gözlem sınıfına ilişkin veriler bulunmaktadır. Bu veriler bir hiper düzlem tarafından ayrılamaz. Şekil 6'da sol taraftaki şekilde gözlemler, aralarında doğrusal olmayan bir sınır bulunan iki sınıfa ayrılabilir. Sağ taraftakinde ise SVM doğrusal bir sınır arar ve sonuç olarak çok düşük performans göstermektedir. Şekil 7'de sol taraftaki şekilde doğrusal olmayan verilere 3'üncü dereceden polinom çekirdeğe sahip SVM uygulanarak çok daha uygun bir karar kuralı elde edilir. Sağ taraftakinde ise radyal çekirdeğe sahip bir SVM uygulanmıştır. Bu örnekte, her iki çekirdek de karar sınırını yakalama yeteneğine sahiptir.

### 1.5.5. Yapay Sinir Ağları

Yapay Sinir Ağları (Artificial Neural Networks-ANN), insan beyninden esinlenilerek geliştirilmiş, biyolojik sinir ağlarını taklit eden bilgisayar programlarıdır (Elmas, 2010: 23). İnsan sinir sistemi, nöron adı verilen hücrelerden oluşur. Biyolojik nöronlar, sinaps adı verilen temas noktalarında birbirine bağlanır. Öğrenme canlı organizmalarda nöronlar arasındaki sinaptik bağlantıların gücünü değiştirerek gerçekleştirilir. Genellikle, bu bağlantıların gücü dış uyaranlara cevap olarak değişir. Yapay sinir ağları bu biyolojik sürecin bir simülasyonu olarak düşünülebilir (Aras, 2013: 5).

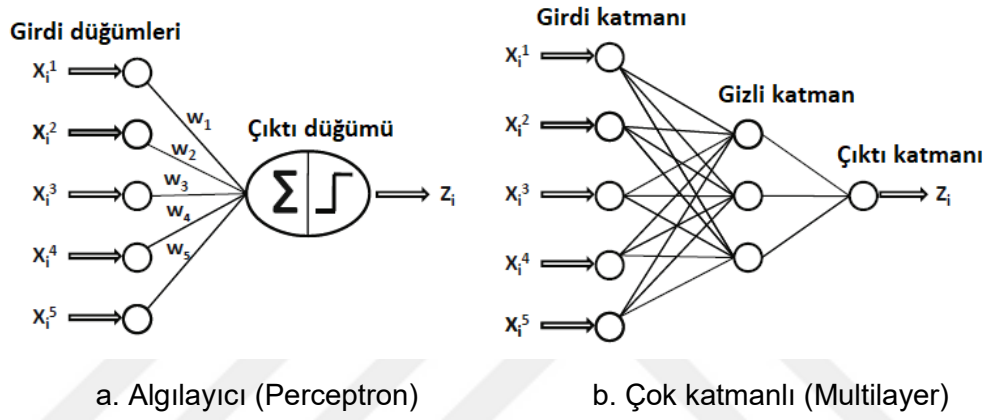
ANN, çeşitli ağırlıklar ile birbirine bağlanmış birçok yapay sinir hücresinden oluşan yapılardır. Girişler  $x_i$  sembolüyle gösterilen ve sinir hücresine dışarıdan gelen bilgilerdir. Bu bilgiler sinir hücresine dışarıdan veya aktif sinirden önce gelen diğer yapay nöronlardan alınabilir. Yapay sinir hücresi içerisindeki girişler ağ üzerin aynı etkiyi vermemekte, hepsinin etkisi farklı olmaktadır. Bu etkinin derecesi,  $w_i$  sembolüyle gösterilen ağırlık değerleri ile belirlenmektedir (Elmas, 2010: 30-31).

Birleştirme/toplama fonksiyonunda,  $x_i$  girişlerinin her biri  $w_i$  ağırlıkları ile çarpılır. Bu girişlerin hepsinin ağırlıklarla çarpılmış hali eşik değeri  $T$  toplanır. Toplama fonksiyonu ile elde edilen değer sonuç elde etmek amacıyla aktivasyon fonksiyonuna gönderilir. Sinir, aktivasyon fonksiyonunun eşik seviyesini geçerse çıktı üretilir. Bu durumun, aktivasyon fonksiyonu 1 ve 0 çıkış değerleriyle çıktı verecek şekilde bir örneği aşağıda verilmiştir (Elmas, 2010: 32-33).

$$y_i = \begin{cases} 1 & \text{eğer } w_1 x_1 + w_2 x_2 + \dots + w_n x_n \geq T \\ 0 & \text{eğer } w_1 x_1 + w_2 x_2 + \dots + w_n x_n < T \end{cases}$$

Sinir ağlarının en temel yapısı sadece bir girdi ve bir çıktı katmanı içeren algılayıcı (perceptron) olarak adlandırılır. Algılayıcı yapısının bir örneği Şekil 8.a'da gösterilmiştir. Girdi katmanları, girilen değerlere gerçekte herhangi bir matematiksel işlem uygulamadan sadece özellik iletir. Bu yüzden algılayıcı modelin öğrenme fonksiyonu yalnızca tek bir çıktı düğümüne dayanan basit bir doğrusal modeldir. Çok katmanlı sinir ağları ise giriş ve çıkış katmanlarına ek olarak bir veya daha fazla gizli katmana sahip ağlardır ve daha karmaşık hesaplamalar içerir (Aggarwal, 2015: 328).

**Şekil 8:** Tek ve Çok Katmanlı Sinir Ağları



Kaynak: Aggarwal, 2015: 328.

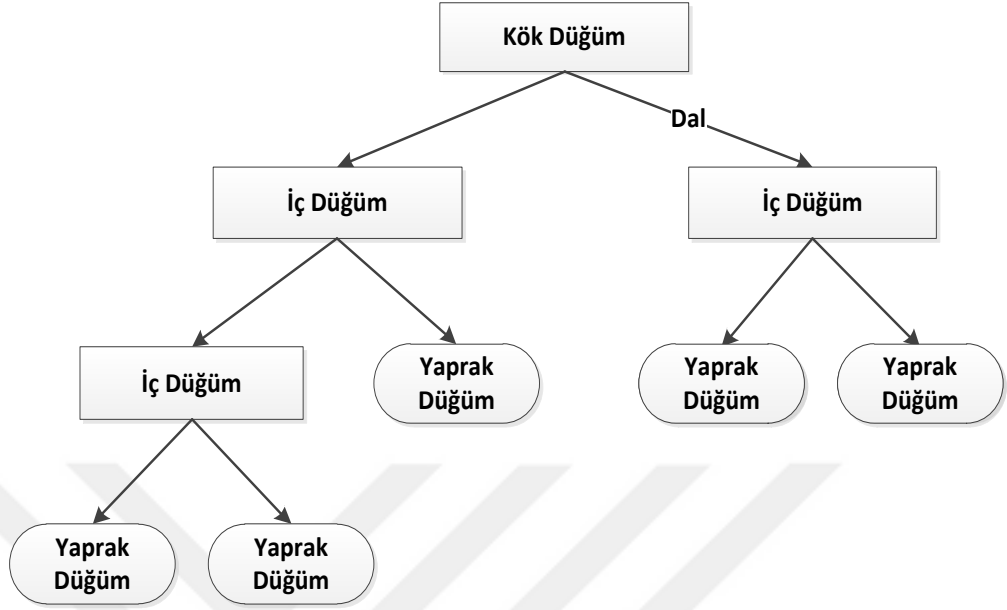
### 1.5.6. Karar Ağaçları

En basit tanımıyla, karar ağacı analizi, sınıflandırmaya ve regresyona yönelik bir böl ve yönet yaklaşımıdır. Mitchell (1997) karar ağacı öğrenimini öğrenilen fonksiyonun bir karar ağacı ile temsil edildiği, kesikli değerli (discrete-valued) hedef fonksiyonlarını tahmin etmek için kullanılan bir yöntem olarak ifade etmiştir. Karar ağacı öğrenmesi, tümevarımsal çıkarım için en yaygın kullanılan ve pratik yöntemlerden biridir (Mitchell, 1997: 54). Karar ağaçları, tahmine dayalı modelleme için büyük veri tabanlarında önemli olan öznitelikleri keşfetmek ve örüntüleri çıkarmak için kullanılabilir. Eğitim kümesinin öznitelik uzayını yinelemeli olarak bölerek bir karar ağacı oluşturulur. Amaç, bilgilendirici ve sağlam bir hiyerarşik sınıflandırma modeli sağlamak için öznitelik uzayını bölen bir dizi karar kuralı bulmaktır. Bir karar kuralı seçildikten sonra, öznitelik uzayı iki ayrık alt uzaya bölünür. Daha sonra, elde edilen tüm alt uzaylar tek bir sınıfa ait örnekler içerene kadar, bölümlenme işlemi sonuçtaki alt uzayların her birine özyinelemeli olarak uygulanır ve ortaya bölünmeleri tanımlayan

bir karar ağacı çıkmaktadır (Myles ve diğerleri, 2004: 276). Karar ağaçları sınıflandırma için bir ağaç oluştur daha sonra, çıktı değeri bilinmeyen girdi verilerinin öznitelik değerleri karar ağacında test edilir. Kökten yaprak düğüme kadar bir yol izlenir ve çıktı için sınıf tahmini yapılır (Mitchell, 1997: 52-53; Kocakoç ve Pulat, 2022: 329-330).

Bir karar ağacı, her bir iç düğümün (yaprak olmayan düğüm) bir öznitelik üzerinde bir testi ifade ettiği, her dalın testin bir sonucunu temsil ettiği ve her yaprak düğümün (veya uç düğümün) bir sınıf etiketine sahip olduğu, akış şeması benzeri bir ağaç yapısıdır (Han ve diğerleri, 2011: 84). Şekil 9' da görüldüğü üzere düğüm, dal ve yaprak olarak adlandırılan üç temel kısımdan oluşur. Üç tür düğüm vardır. (a) Karar düğümü olarak da adlandırılan bir kök düğüm, bir ağaçtaki en üstteki düğümdür. Tüm kayıtların iki veya daha fazla birbirini dışlayan alt kümeye bölünmesiyle sonuçlanacak bir seçimi temsil eder. (b) Şans düğümleri olarak da adlandırılan iç düğümler, ağaç yapısında o noktada mevcut olan olası seçeneklerden birini temsil eder; düğümün üst kenarı üst düğüme, alt kenarı ise alt düğümlerine veya yaprak düğümlerine bağlıdır. (c) Uç düğümler olarak da adlandırılan yaprak düğümler, nihai sonucunu temsil eder. Kök ve yapraklar arasında kalan kısımlar ise dal olarak ifade edilir. Dallar, kök düğümlerden ve iç düğümlerden kaynaklanan tesadüfi sonuçları temsil eder. Dal hiyerarşisi kullanılarak bir karar ağacı modeli oluşturulur. Kökten her bir yaprağa giden tek bir yol veya tek bir karar kuralı vardır. Kök düğümden iç düğümlere kadar olan her yol, bir sınıflandırma karar kuralını temsil eder. Bu karar ağacı yolları, "eğer-o zaman" kuralları olarak da temsil edilebilir. Örneğin, "Koşul 1 ve koşul 2 ve koşul... ve koşul k gerçekleşirse, sonuç j oluşur." (Song ve Lu, 2015: 1-3).

**Şekil 9:** Karar Ağacı Yapısı



Kaynak: Kocakoç ve Pulat, 2022: 330.

Karar ağaçlarının oluşturulmasındaki en önemli adım ağaçtaki dallanmanın hangi kriterle yapılacağı veya hangi öznel değerlerine göre ağaç yapısının oluşturulacağıdır. Aslında her farklı kriter bir karar ağacı algoritmasına karşılık gelmektedir. Bilgi kazancı, kazanç oranı, gini indeksi ve ki-kare istatistiği yaklaşımları başlıca kullanılan dallanma kriterleridir (Kavzoğlu ve diğerleri, 2010: 40). Bu dallanma kriterlerine karar ağacı algoritmalarının içerisinde değinilecektir.

Karar ağaçlarının güçlü yönleri aşağıdaki gibi özetlenebilir (Han ve diğerleri, 2011: 331; Song ve Lu, 2015: 5):

- Karar ağacı sınıflandırıcılarının oluşturulması, herhangi bir alan bilgisi veya parametre ayarı gerektirmeyen parametrik olmayan bir yaklaşımdır.
- Karar ağaçları anlaşılabilir kurallar üretirler.
- Karar ağaçlarının, diğer makine öğrenmesi algoritmalarına göre insanın karar verme sürecini daha fazla yansıtmaktadır
- Edinilen bilgi ağaç biçiminde temsil edilebildiğinden uzman olmayanlar tarafından bile kolaylıkla yorumlanabilir (özellikle küçüklerse).
- Karar ağaçları aşırı hesaplama gerektirmeden sınıflandırma yaparlar.
- Karar ağaçları hem nicel ve hem de nitel değişkenler için uygundur.
- Aykırı değerlere karşı sağlamdır (robust).

- Karar ağaçları çok boyutlu verileri işleyebilir.
- Genel olarak, karar ağacı sınıflandırıcıları iyi bir doğruluğa sahiptir.

Karar ağaçlarının dezavantajı ise; karar ağaçlarında verilerdeki küçük bir değişiklik nihai tahmin edilen ağaçta büyük bir değişikliğe neden olabilmektedir (James ve diğerleri, 2013: 316).

ID3, C4.5, C5.0 ve CART algoritmaları tez kapsamında kullanılan, literatürde bulunan karar ağacı algoritmalarına aşağıda değinilmiştir.

ID3: J. Ross Quinlan (1986) tarafından kategorik öznitelikler için önerilen entropi tabanlı basit bir karar öğrenme algoritmasıdır. ID3 algoritmasının temel fikri, her düğümde her bir özneliği test etmek için verilen eğitim verisi setlerinde yukarıdan aşağıya, açgözlü bir arama yaparak karar ağacını oluşturmaktır. ID3, bölme özneliğini (splitting attribute) seçmek için bilgi kazancı ölçütünü kullanır. Bilgi kazancının hesaplanmasında entropi kavramından yararlanılmaktadır. Entropi kavramı, bir veri kümesi içerisinde belirsizlik ve rastgeleliği ölçmek için kullanılır. Bir özelliğin entropi değeri ne kadar fazla ise, o özellik ile verilecek kararlar o kadar belirsiz ve kararsız olacaktır. Bilgi kazanımı ise, belirli bir özelliğin eğitim örneklerini hedef sınıflandırmalarına göre ne kadar iyi ayırdığını ölçer. Böylece, mevcut düğümün test özneliği olarak en yüksek bilgi kazancına (information gain) (maksimum düzeyde entropi azalması) sahip özneliği seçebiliriz (Priyam ve diğerleri, 2013: 233; Brijain ve diğerleri, 2014: 1-2; Bahety, 2014: 2-3; Niuniu ve Yuxun, 2010: 105-106; Kocakoç ve Pulat, 2022: 334-335).

Entropi, bir sistemdeki düzensizliğin ve belirsizliğin ölçüsüdür ve bir veri kümesinin entropisi, veri kümesinin hedef özelliğindeki düzensizliğin ölçüsüdür. İkili sınıflandırma durumunda (hedef sütunun yalnızca iki sınıfa sahip olduğu durumlarda), hedef sütundaki tüm değerler homojen(benzer) ise entropi 0'dır ve hedef sütun her iki sınıf için eşit sayı değerlerine sahipse 1 olacaktır. Entropi aşağıdaki şekilde hesaplanmaktadır:

$$H(T) = - \sum_{i=1}^n p_i \log_2(p_i)$$

Burada n, hedef sütundaki toplam sınıf sayısıdır ve p<sub>i</sub>, 'i' sınıfının olasılığı veya veri kümesindeki hedef sütunda i sınıfı olan satır sayısının, toplam satır sayısına oranıdır.

#### ID3 Algoritmasının Adımları:

Adım1: Sınıf niteliği için entropi hesaplanır.

$$H(T) = - \sum_{i=1}^n p_i \log_2(p_i)$$

Adım2: Her öznitelik için sınıf niteliği göz önüne alınarak ağırlıklı ortalamalar hesaplanır.

$$H(X,T) = \sum_{i=1}^n \frac{|T_i|}{|T|} H(T_i)$$

Adım3: Her öznitelik için bilgi kazancı hesaplanır.

$$\text{Bilgi Kazancı}(X,T) = H(T) - H(X,T)$$

Adım4: En büyük bilgi kazancı sağlayan öznitelik seçilir.

Adım5: Bir önceki düğümden çıkan her bir dal için veri kümesi yeniden düzenlenir ve Adım1'e dönülür. Her adımda veri kümesi satırları azalır. İşlenecek satır kalmayınca dek işlemlere devam edilir.

C4.5: 1993 yılında Quinlan Ross tarafından hem sürekli hem de kategorik öznitelikler için geliştirilen entropi tabanlı bir algoritmadır. ID3 algoritmasındaki eksik yönleri giderilerek geliştirilmiştir. Bu yöntemde karar ağacında hangi niteliğe göre dallanmanın yapılacağını belirlemek üzere kazanç oranı (gain ratio) ölçütünü kullanır (Priyam ve diğerleri, 2013: 233).

Bilgi kazancı ölçütü, çok sayıda değere sahip öznitelikleri seçmeyi tercih etmektedir. Yani, dallanma yapılacak özniteliği seçerken çok sayıda değere sahip öznitelikleri seçmeye yönelik bir eğilim gösterir. ID3'ün uzantısı olan C4.5, bu öznitelik yanlılığın üstesinden gelmeye çalışan kazanç oranını kullanır (Han ve diğerleri, 2011: 340-341).

C4.5 Algoritmasının Adımları:

Adım1: Sınıf niteliği için entropi hesaplanır.

$$H(T) = - \sum_{i=1}^n p_i \log_2(p_i)$$

Adım2: Her öznitelik için sınıf niteliği göz önüne alınarak ağırlıklı ortalamalar hesaplanır.

$$H(X,T) = \sum_{i=1}^n \frac{|T_i|}{|T|} H(T_i)$$

Adım3: Her öznitelik için bilgi kazancı hesaplanır.

$$\text{Bilgi Kazancı}(X,T) = H(T) - H(X,T)$$

Adım4: Ayırma Bilgisi hesaplanır.

$$H(P_{X,T}) = - \sum_{i=1}^k \frac{|T_i|}{|T|} \log_2 \left( \frac{|T_i|}{|T|} \right)$$

Adım5: Kazanç oranı hesaplanır.

$$\text{Kazanç Oranı}(X,T) = \frac{\text{Bilgi Kazancı}(X,T)}{H(P_{X,T})}$$

Adım6: En yüksek kazanç oranı değerine sahip öznitelik kök düğüm olarak seçilir. Bu öznitelik kendi kategori sayısı kadar dala ayrılarak alt düğümler oluşturulur. Dallanma yapılamayacak düğümler yaprak düğüm olarak etiketlenir.

Adım7: Bir önceki düğümden çıkan her bir dal için veri kümesi yeniden düzenlenir ve Adım1'e dönülür. Ağacın alt düğümleri, yaprak düğüm olana dek süreç devam edilir.

C5.0: C5.0 algoritması, Ross Quinlan tarafından C4.5'e dayalı olarak geliştirilmiştir. Özellikle büyük veri setleri için kullanılan sınıflandırma algoritmasıdır. Hız, bellek ve etkinlik açısından C4.5'ten daha iyidir. C5.0, dallanma kriteri olarak en yüksek bilgi kazanım oranına sahip özniteliği seçer ve daha sonra özyineleme sürecini kullanarak bir ağaç oluşturur (Brijain ve diğerleri, 2014: 2; Zhang ve diğerleri, 2021: 2001).

C5.0 karar ağacının ürettiği kurallar, görülmeyen yani test veri setinde düşük hata oranlarına sahiptir. Dolayısıyla C4.5 ile karşılaştırıldığında, C5.0 algoritmasından daha iyi bir performansa sahiptir. C5.0 otomatik olarak gereksiz özniteliklerin kaldırılmasına izin verir, C5.0 algoritmasında eksik öznitelik değerlerine sahip veriler işlenebilir (Pandya ve Pandya, 2015: 18-19). Artırma tekniği kullanması gibi çeşitli güçlü yönleri vardır. Bu nedenle basit ve yüksek performansa sahip bir karar ağacı oluşturabilir (Zhang ve diğerleri, 2021: 2001).

CART: Sınıflandırma ve regresyon ağaçları anlamına gelir ve Breiman tarafından 1984 yılında geliştirilmiştir. CART tarafından yapılan sınıflandırma ağacı yapısı, özniteliklerin ikili özyinelemeli olarak bölünmesine dayanmaktadır. Yani her dal ikili alt dallara bölünerek büyümektedir (Priyam ve diğerleri, 2013: 233). Hem kategorik hem de sürekli öznitelikleri işleyebilmektedir. Bu yöntemde karar ağacında hangi niteliğe göre dallanmanın yapılacağına Gini katsayısı kullanılarak karar verilir. Karar ağacının karmaşıklığı belirli bir dereceyi aştığında, karmaşıklığın artmasıyla sınıflandırma doğruluğu azalacaktır. Algoritma, maliyet karmaşıklığı budama algoritmasını kullanan son nesil bir ikili ağaçtır (Niuniu ve Yuxun, 2010: 106).

CART Algoritmasının Adımları:

Adım1: Her öznitelik değerleri ikili olacak biçimde gruplanır. Bu şekilde elde edilen sol ve sağ bölümlere karşılık gelen sınıf değerleri gruplandırılır.

Adım2: Her bir öznelikle ilgili sol ve sağ taraftaki bölünmeler için  $Gini_{sol}$  ve  $Gini_{sağ}$  değerleri hesaplanır.

$$Gini_{sol} = 1 - \sum_{i=1}^k \left( \frac{L_i}{|T_{sol}|} \right)^2$$

$$Gini_{sağ} = 1 - \sum_{i=1}^k \left( \frac{R_i}{|T_{sağ}|} \right)^2$$

Adım3: Her  $j$  niteliği için,  $n$  eğitim kümesindeki satır sayısı olmak üzere aşağıdaki değer hesaplanır:

$$Gini_j = \frac{1}{n} (|T_{sol}|Gini_{sol} + |T_{sağ}|Gini_{sağ})$$

Adım4: Her  $j$  niteliği için hesaplanan  $Gini_j$  değerleri arasından en küçük olanı seçilir ve bölünme bu öznelik üzerinden gerçekleştirilir. Bu öznelik kendi kategori sayısı kadar dala ayrılarak alt düğümler oluşturulur. Dalların yapılamayacak düğümler yaprak düğüm olarak etiketlenir.

Adım5: Bir önceki düğümden çıkan her bir dal için veri kümesi yeniden düzenlenir ve Adım1'e dönülür. Ağacın alt düğümleri, yaprak düğüm olana dek süreç devam edilir (Özkan, 2016: 111-112).

### 1.5.7. Topluluk Öğrenmesi Yöntemleri

Sınıflandırma, kümeleme ve regresyon problemlerinde, tek bir öğrenme algoritmasının sahip olduğu kararlılığı ve tahmin doğruluğunu arttırmak için, topluluk öğrenmesi (ensemble learning) kavramı ortaya atılmıştır (Kapucu ve Çubukçu, 2019: 87). Topluluk öğrenmesi yöntemleri, temel öğrenme algoritmalarının genelleştirme hatasını azaltarak ve doğru sınıflandırma oranını artırarak daha yüksek başarımlı tahmin edebilen modellerin oluşturulması amaçlanır (Onan, 2018: 366-367). Topluluk öğrenmesi modeli, bireysel olarak eğitilmiş bir dizi temel modeli belirli bir toplam kuralı kullanarak birleştirerek tek bir nihai tahmin üreten güncel bir makine öğrenmesi araştırma alanıdır. Baz (base) ya da temel modellerin doğru ve çeşitli olması toplama sonuçlarının bireysel modellere göre daha doğru sonuçlar vermesini sağlamaktadır. Topluluk modeli, geleneksel bireysel tahmin modelinden daha güvenilir ve doğru tahminler sağlayabilmektedir (Verma ve Mehta, 2017: 155).

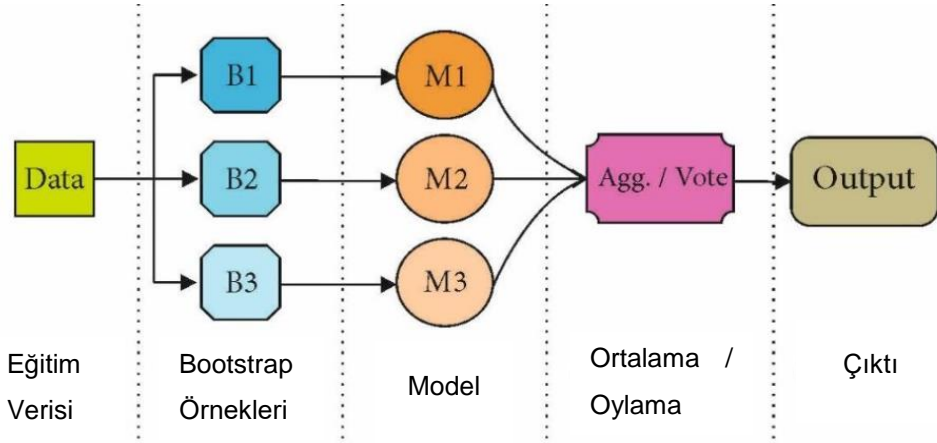
Topluluk öğrenmesi yöntemi, temel model oluşturma stratejisine göre heterojen topluluk ve homojen topluluk olmak üzere iki kategoriye ayrılabilir. Heterojen topluluk modelinde; temel modelleri, aynı eğitim verilerini farklı öğrenme algoritmalarına veya aynı algoritmalara farklı parametre ayarlarıyla uygulanarak oluşturulur. Buna örnek olarak yığın (stacking) verilebilir. Homojen topluluk modelinde ise, orijinal verilerden yeniden örneklenen farklı eğitim verilerini aynı parametre ayarları ile aynı öğrenme algoritmasına uygulayarak temel modellerini oluşturur. Buna örnek olarak torbalama, artırma, rasgele orman (random forest), rastgele alt uzay (random subspace) verilebilir (Wang ve diğerleri, 2018: 110). Böylece, heterojen topluluk modeli farklı öğrenme algoritmalarının birbirlerini tamamlamasını sağlayarak, homojen topluluk modeli ise temel olarak seçilen bir öğrenme algoritmasını, eğitim verilerinin farklı dağılımlarıyla eğiterek tahmin doğruluklarını artırırlar (Kapucu ve Çubukçu, 2019: 87).

#### **1.5.7.1. Torbalama**

Bootstrap Aggregating olarak da bilinen Bagging algoritması, makine öğrenimi algoritmalarının doğruluğunu artıran, varyansı azaltan ve aşırı uyumu önleyen topluluk öğrenmesi yöntemidir. Torbalama (Bagging) yönteminde, eğitim kümesinden bootstrap örnekler üzerinden birden fazla sınıflandırıcının eğitilmesine dayanmaktadır. Bootstrap uygulaması, iadeli rastgele seçim yaparak alt örneklemeler oluşturmak için kullanılmaktadır.

Torbalama algoritmasında  $n$  adet örnek içeren eğitim kümesinden, yine  $n$  adet örneğe sahip bir eğitim seti rastgele yerine koyarak türetilir. Bazı örnekler yeni eğitim kümesinde yer almazken bazıları birden fazla defa yer alabilmektedir. Daha sonra farklı örnekler içeren eğitim setleri ile modeller eğitilir ve modellerin sonuçları sınıflandırma probleminde oylama; regresyon probleminde ortalaması alınarak birleştirilip tek bir çıktı elde edilmektedir. Torbalama algoritmasına ilişkin şekil aşağıda verilmiştir. En sık kullanılan torbalama yöntemlerinden biri rasgele orman algoritmasıdır.

**Şekil 10:** Torbalama Algoritması

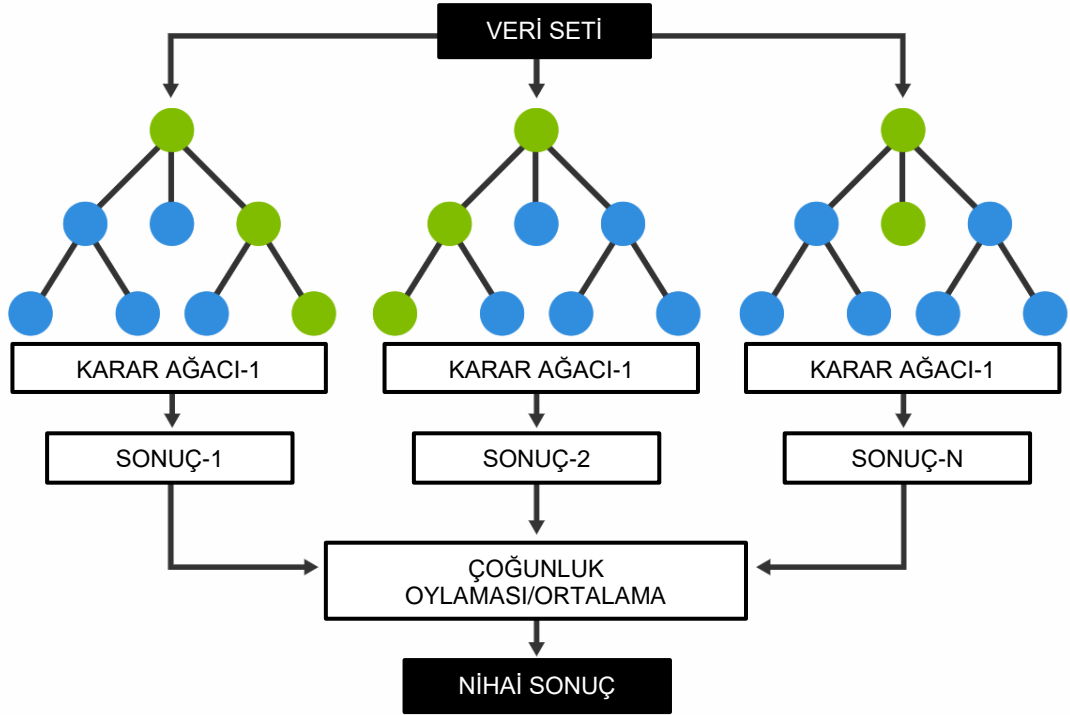


Kaynak: Nazlı, 2021: 1.

Rasgele Orman (Random Forest-RF): Breiman 2001 yılında torbalama algoritmasına bir rastgelelik katmanı ekleyerek rasgele orman yöntemini önermiştir. Topluluk sınıflandırma algoritmaları içinde en yaygın kullanılan RF algoritması, içinde oylama metodunu barındıran bir algoritmadır. Standart ağaçlarda, her bir düğüm tüm değişkenler arasında en iyi olanı kullanılarak dallandırma yapılırken rasgele ormanda ise, her bir düğümde rasgele seçilen değişkenleri kullanarak her bir düğüm dallara ayrılmaktadır (Liaw ve Wiener, 2002: 18). Karar ağacı oluşturulmasında CART yöntemi kullanılır yani dallanma kriteri olarak da en yaygın kullanılan ölçüt Gini indeksidir. Rasgele oluşturulmuş birden fazla karar ağacı modeli eğitildikten sonra; modellerin sonuçları sınıflandırma probleminde oylama, regresyon probleminde ortalaması alınarak birleştirilip tek bir çıktı elde edilmektedir.

RF yöntemi, oluşturulacak olan ağaç sayısı ve her düğüm ayırımında rasgele seçilecek olan değişken sayısı olmak üzere sadece iki parametreye sahiptir. Bu bakımdan kullanıcı dostudur ve genellikle parametrelerin farklı değerlerine karşı çok hassas değildir. Bu yöntem diğer birçok sınıflandırıcıya göre daha iyi performans göstermektedir ve aşırı uyum (overfitting) probleminin önüne geçebildiğinden güçlü (robust) bir metottur (Liaw ve Wiener, 2002: 18). RF Algoritmasına ilişkin şekil aşağıda verilmiştir.

Şekil 11: Rasgele Orman Algoritması



Kaynak: TIBCO, <https://www.tibco.com/reference-center/what-is-a-random-forest>, 15.05.2023.

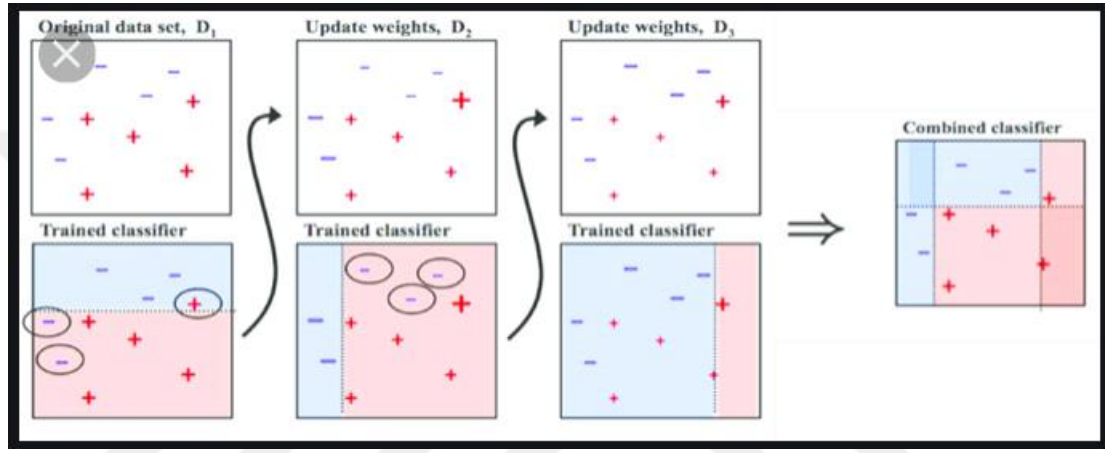
### 1.5.7.2. Artırma

Artırma (Boosting), karar ağacının tahmin sonuçlarını iyileştirmenin başka bir yoludur. Daha önce değinilen torbalama gibi, artırma de hem regresyon hem de sınıflandırma problemleri için farklı istatistiksel öğrenme yöntemlerine uygulanabilmektedir. Torbalama algoritmasında, bootstrap örnekler üzerinden birden fazla sınıflandırıcı eğitilir ve son olarak da tüm bu uygun ağaçları tek bir model oluşturmak için birleştirmektedir. Modeller paralel olarak çalışır ve birbirinden bağımsızdır. Artırma algoritmasında ise eğitilen modellerin her biri daha önce eğitilmiş modellerden gelen bilgiler kullanılarak büyütülür. Bu süreç sıralı büyüme olarak adlandırılabilir. Artırma yönteminin ana fikri; modelleri sırayla eğitmektir (Liaw ve Wiener, 2002: 18).

İlk olarak, veri noktalarının her birine eşit ağırlık atanır ve bu modele girdi olarak sağlanarak model eğitilir ve yanlış sınıflandırılmış veri noktalarını belirlenir.

Daha sonra yanlış sınıflandırılmış veri noktalarının ağırlığını artırılarak model eğitilir. İkinci model, ağırlıkları artırılmış veriler üzerinden çalışmaktadır. O da yanlış sınıflandırdıklarını bir sonraki modele artırılmış ağırlıklarla iletmektedir. Modelin performansı, önceki modelde yanlış sınıflandırılmış örneklere daha yüksek bir ağırlık atanarak iyileştirilebilmektedir. En sık kullanılan artırma yöntemleri AdaBoost ve Gradient Artırma yöntemleridir. Artırma algoritmasına ilişkin şekil aşağıda verilmiştir.

**Şekil 12:** Artırma Algoritması



Kaynak: OPENGENUS, <https://iq.opengenus.org/types-of-boosting-algorithms/>, 15.05.2023.

Adaboost: AdaBoost “Adaptive Boosting” ifadesinin kısaltması olan topluluk öğrenmesi yöntemidir. AdaBoost, bir dizi bireysel sınıflandırıcının yinelemeli olarak üretildiği ve topluluktaki her sınıflandırıcının eğitim verilerini doğru bir şekilde sınıflandırmaya çalıştığı en yaygın kullanılan artırma algoritmalarından biridir. Sınıflandırıcı, eğitim örneklerini seçmek için uyarlanabilir bir yeniden örnekleme (adaptive resampling) tekniği kullanır. Örneğin, önceki model tarafından üretilen yanlış sınıflandırılmış bir veri kümesi, doğru sınıflandırılmış olandan daha sık seçilmektedir, böylece yeni model yeni veri kümesinde daha iyi performans gösterebilmektedir. Her iterasyonda, veri kümesine bir ağırlık atanır, böylece sonrasında daha önce yanlış sınıflandırılmış yeniden ağırlıklandırılmış veri kümelerine odaklanmaktadır. Nihai model, tüm temel modellerinin ağırlıklı toplamından elde edilir. Ayrıca AdaBoost, zayıf öğrenciler tarafından değişkenlerin hangi sıklıkla seçildikleri incelenerek değişkenlerin uygunluğunu değerlendirmek için kullanılabilir (Kadavi ve diğerleri, 2018: 9-10).

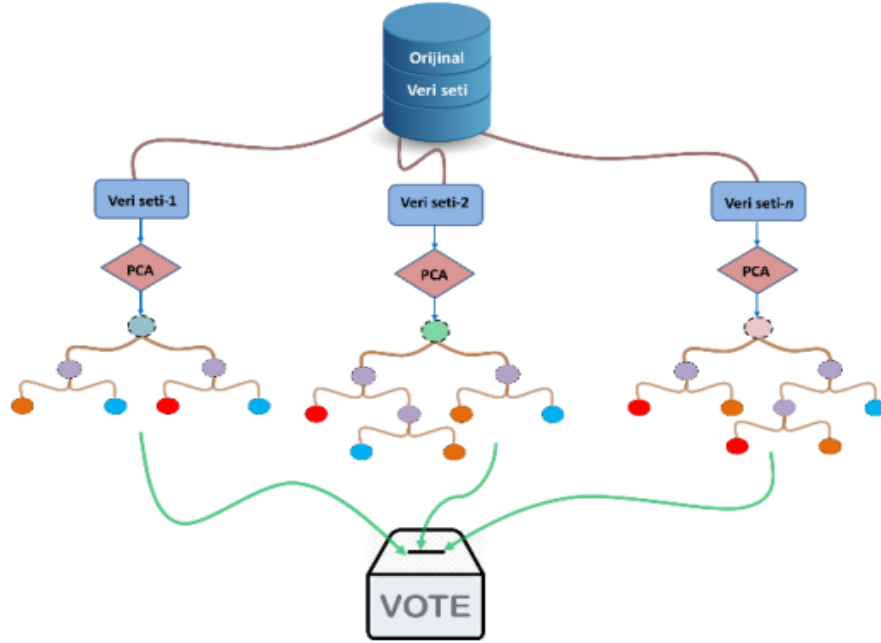
Gradyan Artırma (Gradient Boosting-GB): GB, gradyan iniş yöntemi (gradient descent method) ile hataların minimize edildiği artırma algoritmalarından özel bir çeşiddir. GB, her yeni modelde kayıp fonksiyonunu en aza indirmek amacıyla bir grup zayıf modelden sırayla yeni modeller üretilmektedir yani birden fazla model sırayla eğitilmektedir. Kayıp fonksiyonunu hesaplamak için gradyan iniş yöntemi kullanılır ve bu nedenle, her yeni model kayıp fonksiyonunun kullanımıyla daha doğru bir şekilde fit edilmektedir. Böylece, modelin performansı artmaktadır. Öte yandan, artırma bir noktada durdurulmalıdır; aksi takdirde, model de aşırı uyum problemi ortaya çıkabilmektedir (Nishat ve diğerleri, 2021: 3).

Artırma, hesaplama zamanı açısından torbalama algoritmasına göre çok daha yavaştır. Öte yandan, çoğu durumda artırma, torbalama algoritmasında önemli ölçüde daha iyi bir performansa sahiptir (Briem ve diğerleri, 2002: 2293).

### **1.5.7.3. Rotasyon Ormanı**

Rotasyon ormanı (Rotation Forest-RoF), öznitelik çıkarımına dayalı sınıflandırıcı toplulukları oluşturan son yıllarda kullanılan yeni nesil topluluk öğrenmesi yöntemidir. İlk olarak, öznitelik kümesi rastgele  $n$  alt kümeye bölünür ( $n$ , algoritmanın bir parametresidir) ve her alt kümeye Temel Bileşen Analizi (Principal Component Analysis-PCA) uygulanarak öznitelik çıkarımı yapılır. Daha sonra her alt kümede bir karar ağacı modeli eğitilir ve modellerin sonuçları oylama ile birleştirilerek tek bir çıktı elde edilmektedir. Şekil 13' te RoF algoritmasının çalışması verilmiştir (Witten ve diğerleri, 2005: 357-358; Rodriguez ve diğerleri, 2006: 1620-1621).

**Şekil 13:** Rotasyon Ormanı Algoritması



Kaynak: Çölkesen ve Ertekin, 218: 4.

#### 1.5.7.4. Yığın – İstifleme

Yığın (stacking) algoritması, farklı sınıflandırıcı modeller kullanılarak oluşturulan heterojen topluluk öğrenmesi yöntemidir. Yakın zamanda geliştirilmiş olmasına rağmen, hangi modeller kullanılarak daha iyi performans elde edileceğine ilişkin kabul edilen bir yol olmadığından yani deneme yanılma ile kullanılacak modeller belirlendiğinden torbalama ve artırma yöntemlerinden daha az kullanılmaktadır. Torbalama ve artırma algoritmaları aynı modelleri birleştirmek için kullanılırken; yığın algoritmasında farklı olarak aynı tipteki modeller birleştirmek için kullanılmaz. Bunun yerine, farklı öğrenme algoritmaları tarafından oluşturulan modellere uygulanmaktadır (Witten ve diğerleri, 2005: 369).

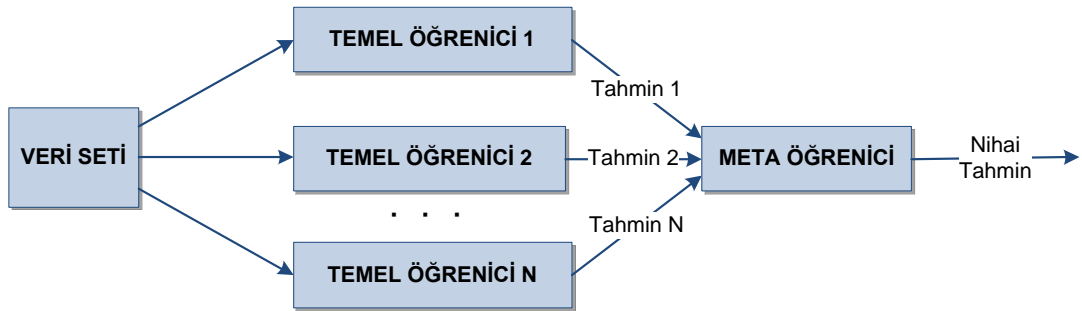
Bireysel modellerde, algoritmalar ayrı ayrı eğitilir ve test verisi üzerinde performansına bakılarak en iyi model seçilir. Fakat daha iyi bir yol yok mu? Denenen algoritmaların tamamını tahmin için kullanarak çıktıları birleştiremez miyiz? Çıktıları birleştirmenin bir yolu, torbalama ya da artırma algoritmasındaki çalışma prensibi olan oylama yaklaşımıdır. Bununla birlikte, oylama, yalnızca temel modeller nispeten iyi performans gösteriyorsa anlamlıdır. Diyelim ki, üç sınıflandırıcı bir torbalama ya da artırma algoritması kullanıyoruz. Üç sınıflandırıcıdan ikisi yanlış tahminlerde bulunursa, oylama sonucu çok kötü bir performans elde ederiz. Bunun yerine, yığın, oylama prosedürünün yerini alan bir meta-öğrenci kavramını ortaya atmıştır.

Oylamayla ilgili problem, hangi sınıflandırıcıya güvenileceğinin net olmamasıdır. Yiğın, temel öğrencilerin çıktılarının en iyi nasıl birleştirileceğini belirlemek için başka bir öğrenme algoritması olan meta öğrenciyi kullanarak çıktıları birleştirme yaklaşımı öne sürülmüştür (Witten ve diğerleri, 2005: 369).

Şekil 14 incelendiğinde, yiğın algoritmasının iki seviyeli bir sınıflandırma yaklaşımı olduğu görülmektedir. Birinci seviyede ikiden fazla temel öğrenci eğitilir, daha sonra birinci seviye öğrencilerin tahminleri ikinci seviye olan meta seviye sınıflandırıcı tarafından birleştirilir. Bu yöntem aşırı uyum (overfitting) riskini azaltmaktadır (Salunkhe ve Mali, 2016: 727-728). Bu algoritmada temel problem, birinci ve ikinci seviyede hangi algoritmaların kullanılacağıdır. Prensipte olarak, herhangi bir öğrenme planı uygulanabilmektedir çünkü en iyi performansı verebilecek algoritmalara ilişkin genel geçer bir kabul bulunmamaktadır. Ancak, yiğın algoritmasını ortaya atan David Wolpert (1992) ikinci seviye sınıflandırıcı olarak basit doğrusal modeller veya yapraklarında doğrusal modeller bulunan ağaçların genellikle iyi çalıştığını öne sürmüştür (Witten ve diğerleri, 2005: 370-371).

Yığılmış genelleme fikrini Wolpert (1992) ortaya atmıştır; Breiman (1996) tarafından sayısal tahmine uygulanmıştır. Ting ve Witten (1997) farklı 2. seviye modelleri (meta öğrenci) deneysel olarak karşılaştırmış ve basit doğrusal modelin en iyi performansı gösterdiğini bulmuşlardır (Witten ve diğerleri, 2005: 372).

**Şekil 14:** Yiğın Algoritması



Kaynak: Yazar tarafından oluşturulmuştur.

## 1.6. SINIFLANDIRICI PERFORMANS DEĞERLENDİRME ÖLÇÜTLERİ

Sınıflandırma modeli elde edildikten sonra bu modelin performansını belirlemek için farklı ölçütler kullanılmaktadır. Bu tür hesaplamalar özellikle farklı modellerin performanslarını kıyaslayabilmek amacıyla yapılmaktadır. Böylece veri

setine en uygun sınıflandırma modelinin hangisi olduğu belirlenmeye çalışılır (Özkan, 2016: 83-84). Bu amaçla model başarısını değerlendirmek için doğruluk, kesinlik, duyarlılık, F-ölçütü en yaygın kullanılan ölçütlerdir. Doğruluk, kesinlik, duyarlılık, F-ölçütünün hesaplanmasında Tablo 1’de verilen karmaşıklık matrisinden (confusion matrix) faydalanılmaktadır. Karmaşıklık matrisinde hedef (çıktı) niteliğinin sınıflarına ait gerçek ile tahmin edilen sayılar bir arada verilmektedir. Karmaşıklık matrisi, gerçek değerler ile tahmin edilen değerleri karşılaştırmak amacıyla kullanılır.

–Gerçek Pozitifler (True Positive-TP): Gerçekte 1 sınıfında olup, sınıflandırıcı tarafından 1 sınıfa atananların sayısını gösterir.

–Gerçek Negatifler (True Negative-TN): Gerçekte 0 sınıfında olup, sınıflandırıcı tarafından 0 sınıfa atananların sayısını gösterir.

–Yanlış Pozitifler (False Positive-FP): Gerçekte 0 sınıfta olup, sınıflandırıcı tarafından 1 sınıfa atananların sayısını gösterir.

–Yanlış Negatifler (False Negative-FN): Gerçekte 1 sınıfta olup, sınıflandırıcı tarafından 0 sınıfa atananların sayısını gösterir.

**Tablo 1:** Karmaşıklık Matrisi (Confusion Matrix)

		Tahmin Edilen Sınıf	
		Sınıf=1	Sınıf=0
Gerçek Sınıf	Sınıf=1	TP	FN
	Sınıf=0	FP	TN

Kaynak: Yazar tarafından oluşturulmuştur.

Doğruluk (Accuracy): Doğru sınıflandırılmış örnek sayısının, toplam örnek sayısına oranıdır ve aşağıdaki gibi hesaplanır:

$$Doğruluk = \frac{TP + TN}{TP + FP + FN + TN}$$

Kesinlik (Precision-Positive Predictive Value): Doğru sınıflandırılmış pozitif örneklerin toplam pozitif tahmin edilen örneklere oranıdır. Bir başka deyişle; sınıfı 1 olarak tahminlenmiş TP örnek sayısının, sınıfı 1 olarak tahminlenmiş tüm örnek sayısına oranıdır. Kesinlik değeri aşağıdaki gibi hesaplanır:

$$Kesinlik = \frac{TP}{TP + FP}$$

Duyarlılık (Sensitivity-Recall-True Positive Rate): Modelin, pozitif sınıf etiketini tahmin etmedeki etkililiğini göstermektedir. Doğru sınıflandırılmış pozitif örnek

sayısının toplam pozitif örnek sayısına oranlanması ile elde edilir. Duyarlılık deęeri ařaęıdaki gibi hesaplanır:

$$\text{Duyarlılık} = \frac{TP}{TP + FN}$$

Belirleyicilik / Özgüllük (Specificity-True Negative Rate): Modelin, negatif sınıf etiketini tahmin etmedeki etkililięini göstermektedir. Doğru sınıflandırılmıř negatif örnek sayısının toplam negatif örnek sayısına oranlanması ile elde edilir. Belirleyicilik deęeri ařaęıdaki gibi hesaplanır:

$$\text{Belirleyicilik} = \frac{TN}{TN + FP}$$

F-Ölçütü (F-Criterion): Kesinlik ve duyarlılık ölçütlerinin harmonik ortalaması alınarak ařaęıdaki gibi hesaplanır:

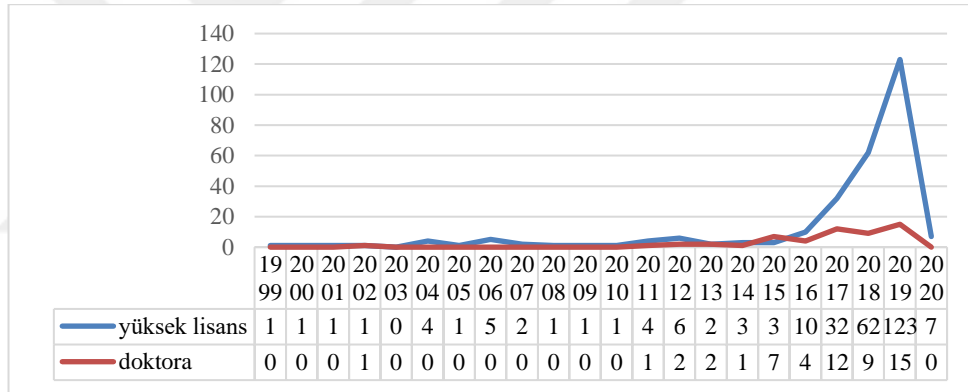
$$F - \text{Ölçütü} = \frac{2 \times \text{Duyarlılık} \times \text{Kesinlik}}{\text{Duyarlılık} + \text{Kesinlik}}$$

## İKİNCİ BÖLÜM LİTERATÜR TARAMASI

### 2.1. TEZ ÇALIŞMALARININ LİTERATÜR İNCELEMESİ

Yüksek Öğretim Kurumu'nun Ulusal Tez Merkezinde "makine öğrenmesi", "machine learning" anahtar kelimeler kullanılarak 2020 yılı mayıs ayında yapılan tarama sonucu 325 teze ulaşılmıştır. Bu tezlerin 25 tanesi erişimi sınırlandırılmıştır. İncelenen tezlerin adlarına <https://tinyurl.com/2mhsaeky> linkinden ulaşabilirsiniz. Tezler; yıl, enstitü, anabilim dalı, kullanılan yöntem, kullanılan anahtar kelimeler dikkate alınarak bibliyografik açıdan detaylı olarak incelenmiştir. Elde edilen veriler Excel programına aktarılarak grafikler oluşturulmuştur (Pulat ve Kocakoç, 2021).

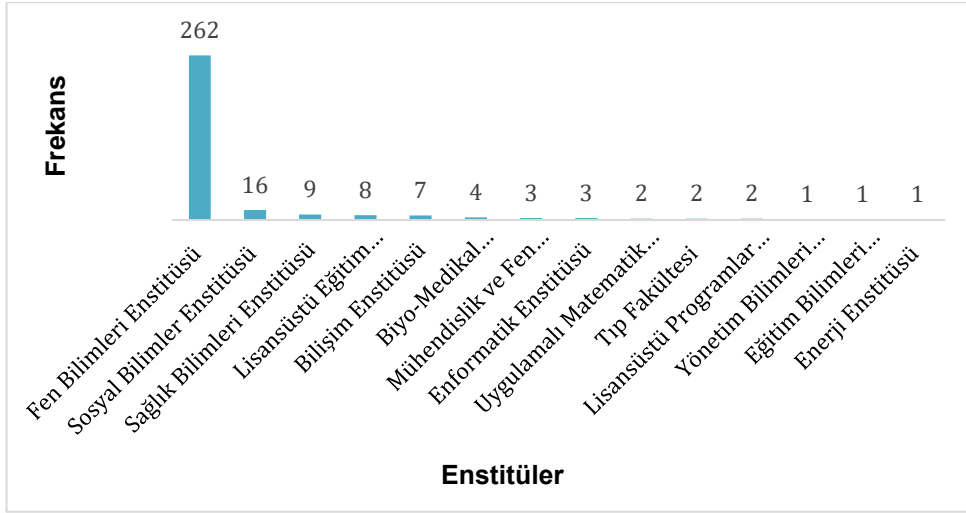
**Şekil 15:** Tezlerin Yıllara göre Dağılımı



Kaynak: Yazar tarafından oluşturulmuştur.

Şekil 15' te verilen lisansüstü tezlerin yıllara göre dağılımı incelendiğinde, 2016 yılına kadar büyük sıçramalar olmayan tez sayılarında 2016 yılından itibaren artışlar gözlenmektedir ve tezlerin çoğunluğunu yüksek lisans tezleri oluşturmaktadır.

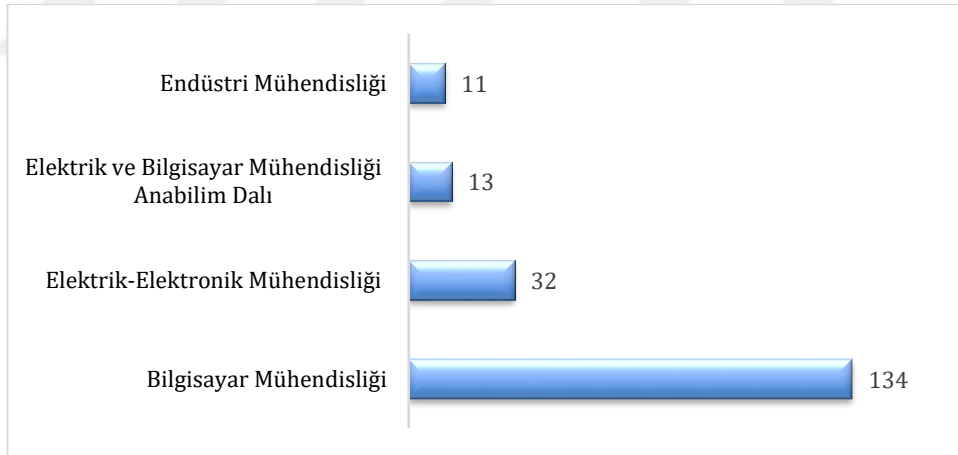
**Şekil 16:** Enstitülere göre Tezlerin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

Şekil 16' da verilen lisansüstü tezlerin enstitülere göre dağılımı incelendiğinde, en çok tezin fen bilimleri enstitüsü altında hazırlandığı görülmektedir.

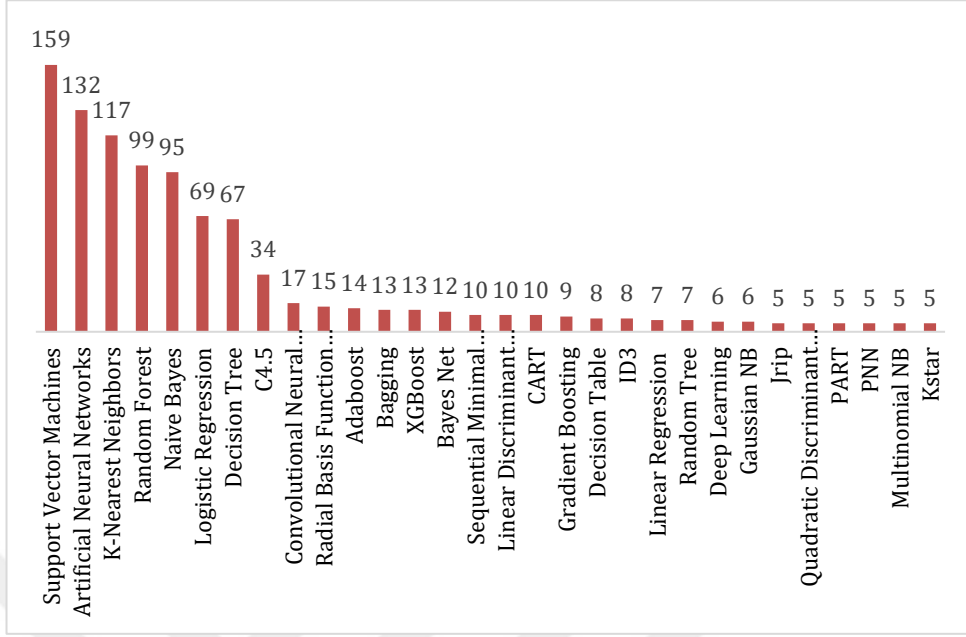
**Şekil 17:** Ana Bilim Dallarına göre Tezlerin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

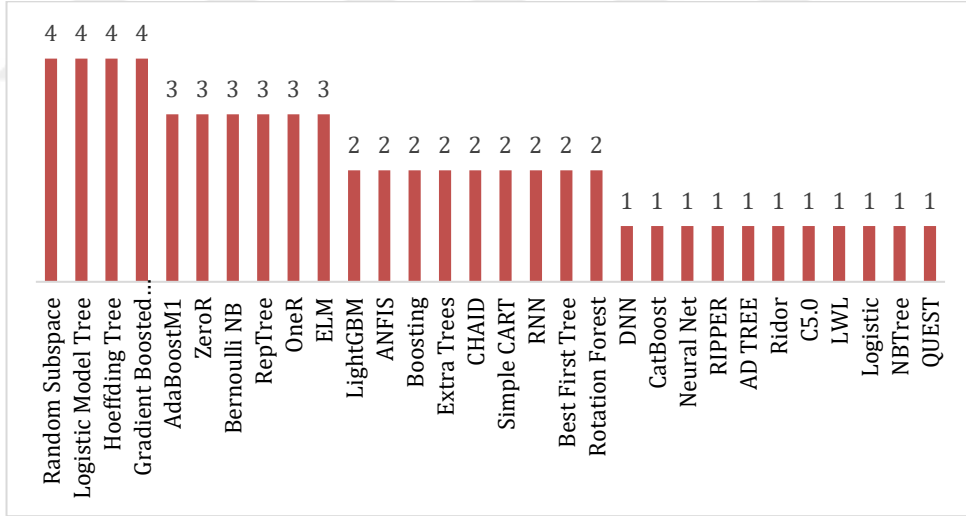
Şekil 17' de tezlerin en çok yayınladığı dört anabilim dalı verilmiştir. Tezlerin büyük bir kısmının Mühendislik alanında yazıldığı görülmektedir.

Şekil 18: 5 Defa ve Üzeri Kullanılan Yöntemlerin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

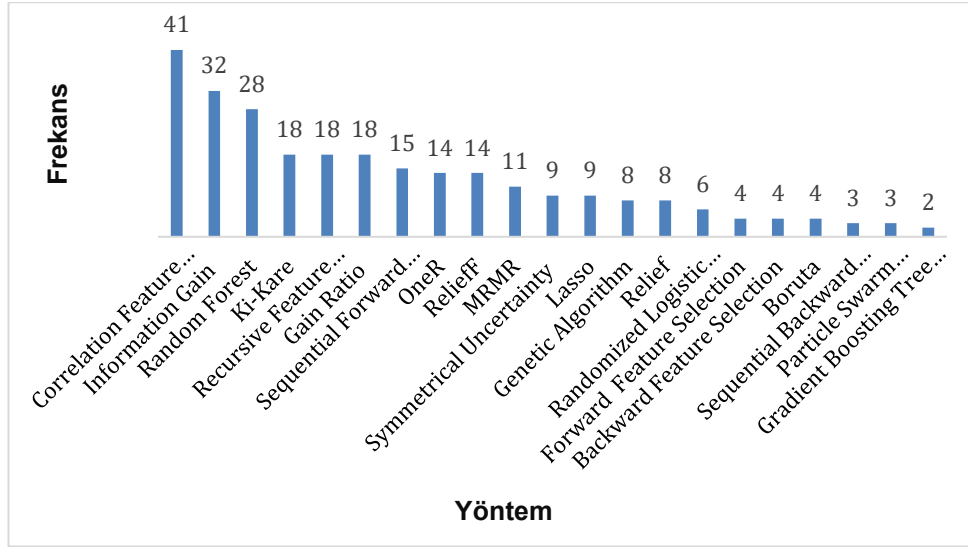
Şekil 19: 5'ten Az Kullanılan Yöntemlerin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

İncelenen tezlerde kullanılan makine öğrenmesi algoritmaları Şekil 18 ve Şekil 19' da verilmiştir. Şekiller incelendiğinde; Destek Vektör Makineleri, Yapay Sinir Ağları, K- en Yakın Komşu, Rasgele Orman, Naive Bayes, Lojistik Regresyon, Karar Ağaçları en sık kullanılan yöntemlerdir. Şekiller incelendiğinde topluluk öğrenmesi yöntemlerinin çok az kullanıldığı görülmektedir.

**Şekil 20:** Tezlerde Kullanılan Öznitelik Seçim Yöntemlerinin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

Kullanılan öznitelik seçim yöntemleri Şekil 20’ de görülmektedir. Grafik incelendiğinde; Korelasyon Tabanlı Öznitelik Seçimi (Corelation Feature Selection), Bilgi Kazancı (Information Gain), Rasgele Orman (Random Forest), Ki-Kare, Özyinelemeli Öznitelik Eliminasyonu (Recursive Feature Elimination), Kazanç Oranı (Gain Ratio) en sık kullanılan öznitelik seçim yöntemleridir. Öznitelik seçim yöntemlerinde çoğunlukla klasik öznitelik seçim yöntemlerinin kullanıldığı görülmektedir

Kullanılan anahtar kelimeler dikkate alınarak R programı ile oluşturulan kelime bulutu Şekil 21’ de verilmiştir. Buna göre, “machine learning” (163 kez), “support vector machines” (39 kez), “artificial neural network” (38 kez), “classification” (33 kez), ve “data mining” (24 kez) en çok kullanılan anahtar kelimelerdir.



Şekil 22' de araştırma kapsamında erişilebilen makalelerin yıllar bazında kullanılan dil de dikkate alınarak dağılımı verilmiştir. Yıllar bazında yayınlanan makaleler incelendiğinde 2008 ve sonrasında yayınlanan makale sayısında artış görülmektedir. Aynı zamanda araştırma kapsamına giren makalelerin çoğunluğunu İngilizce yazılmış makaleler oluşturmaktadır.

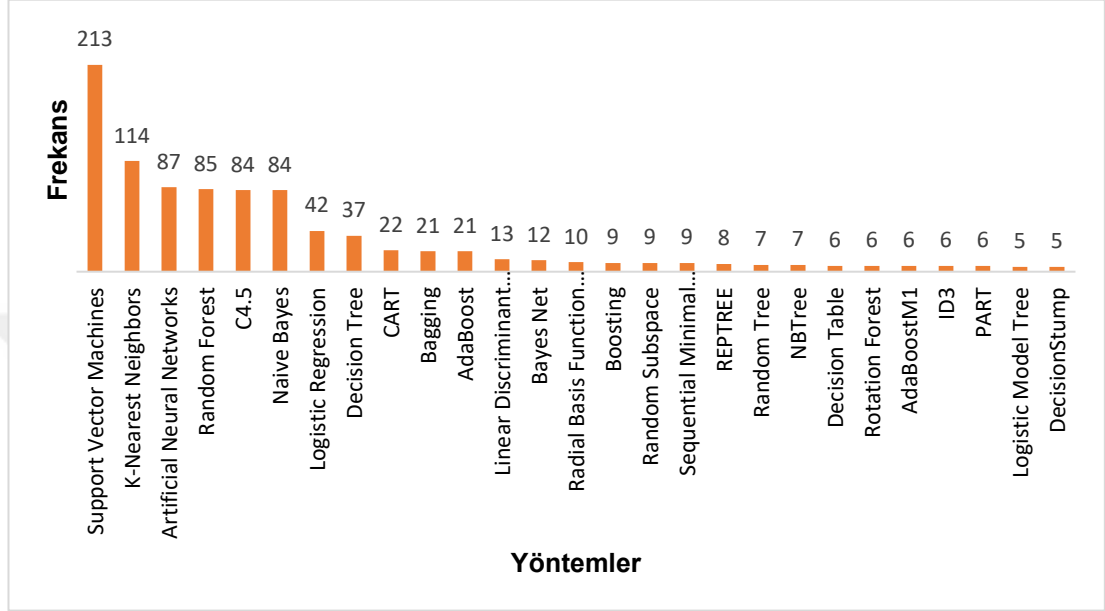
**Tablo 2:** 500 ve Üzeri Atıf Alan Makaleler

Yazar / lar	Yıl	Makale Adı	Atıf Sayısı
Quinlan	1996	Bagging, boosting, and C4. 5	2351
Montana ve Davis	1989	Training Feedforward Neural Networks Using Genetic Algorithms	1724
Huang ve Wang	2006	A GA-based feature selection and parameters optimization for support vector machines.	1635
Lin ve diğerleri	2008	Particle swarm optimization for parameter determination and feature selection of support vector machines	959
Cai ve diğerleri	2018	Feature selection in machine learning: A new perspective	913
Akay	2009	Support vector machines combined with feature selection for breast cancer diagnosis	857
Das	2001	Filters, wrappers and a boosting-based hybrid for feature selection	852
Khalid ve diğerleri	2014	A survey of feature selection and feature extraction techniques in machine learning	817
Patel ve diğerleri	2015	Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques	789
De Mántaras	1991	A distance-based attribute selection measure for decision tree induction	615
Rodriguez-Galiano ve diğerleri	2015	Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines.	585
Huang ve Dun	2008	A distributed PSO–SVM hybrid system with feature selection and parameter optimization	551
Zheng ve diğerleri	2014	Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms.	511
Wen ve diğerleri	2015	A rapid learning algorithm for vehicle classification	509
Uğuz	2011	A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm	507

Kaynak: Yazar tarafından oluşturulmuştur.

Çalışmada incelenen makalelere ilişkin atıf sayıları (2022 Haziran ayı) elde edilmiş ve kaydedilmiştir. Tablo 2' de sadece 500 ve üzeri atıf alan makaleler verilmiştir.

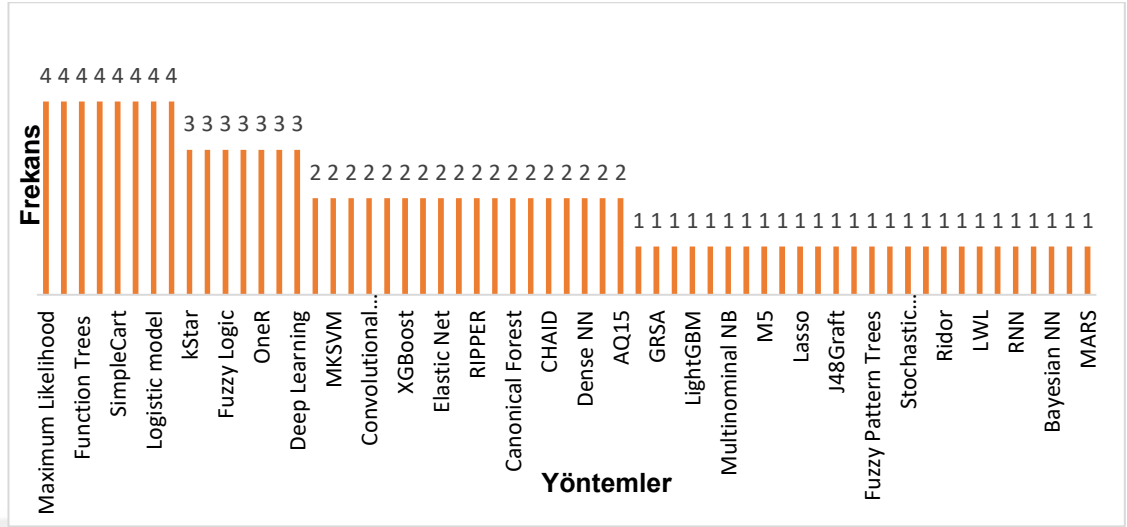
**Şekil 23:** 5 Defa ve Üzeri Kullanılan Yöntemlerin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

Makine öğrenmesi alanında çok sayıda algoritma mevcuttur. Beş defa ve üzeri kullanılan yöntemler Şekil 23'te görülmektedir. Buna göre en sık kullanılan altı yöntem: Destek Vektör Makineleri (Support Vector Machines), K- en Yakın Komşu (K-Nearest Neighbors), Yapay Sinir Ağları (Artificial Neural Networks), Rasgele Orman (Random Forest), C4.5 ve Naive Bayes yöntemleridir. Beşten az kullanılan yöntemler ise Şekil 24' te verilmiştir.

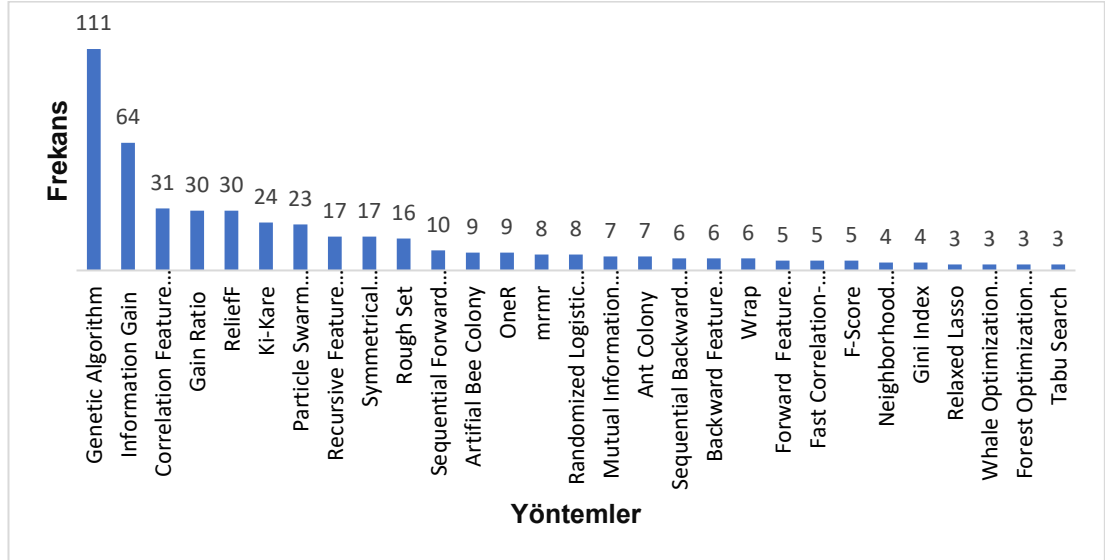
**Şekil 24:** 5'ten Az Kullanılan Yöntemlerin Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

Çalışmalarda kullanılan yöntemlere bakıldığında topluluk öğrenme yöntemlerinin (Bagging, Boosting, AdaBoost, Rotation Forest, AdaBoostM1, XGBoost, LightGBM) yeterince kullanılmadığı bu alanda önemli bir boşluk olduğu görülmektedir.

**Şekil 25:** Makalelerde 2' den Fazla Kullanılan Öznitelik Seçim Yöntemlerinin Frekansı

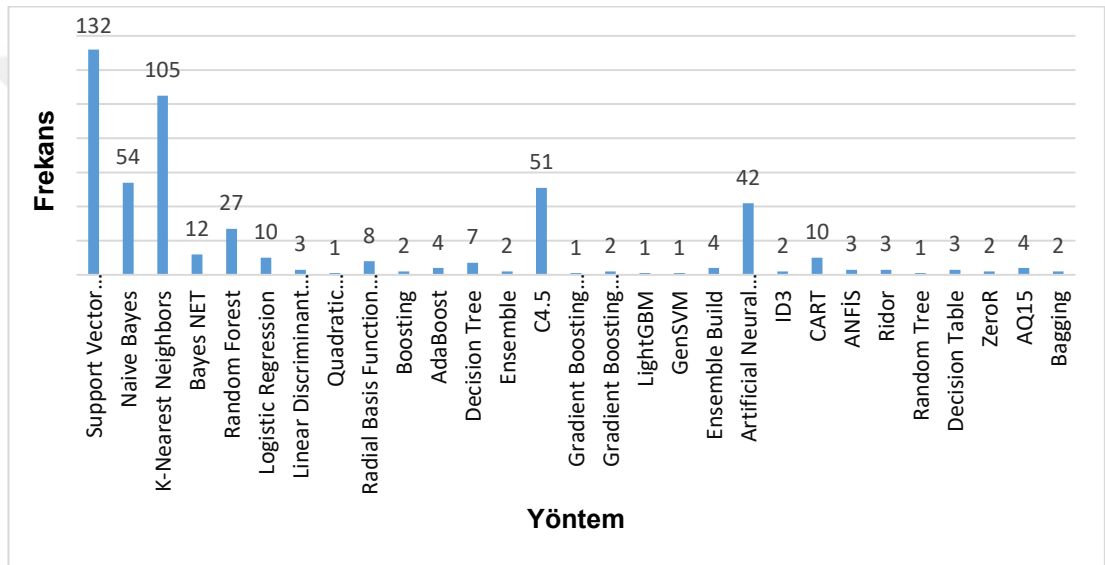


Kaynak: Yazar tarafından oluşturulmuştur.

İncelenen makalelerde kullanılan öznitelik seçim yöntemleri Şekil 25' te görülmektedir. Grafik incelendiğinde; Genetik Algoritma (Genetic Algorithm-GA), Bilgi

Kazancı (Information Gain-IG), Korelasyon Tabanlı Öznitelik Seçimi (Correlation Feature Selection-CFS), Kazanç Oranı (Gain Ratio-GR), ReliefF, Ki-Kare, Parçacık Sürü Optimizasyonu (Particle Swarm Optimization-PSO) en sık kullanılan öznitelik seçim yöntemleridir. Öznitelik seçim yöntemlerinde (GA ve PSO hariç) çoğunlukla klasik öznitelik seçim yöntemleri kullanılmaktadır. Öznitelik seçim yöntemlerinin hangi makine öğrenme algoritmalarıyla birlikte kullanıldığını gösteren grafik aşağıda verilmiştir.

**Şekil 26:** Makalelerde Öznitelik Seçimi ile Birlikte Kullanılan Makine Öğrenme Algoritmalarının Frekansı



Kaynak: Yazar tarafından oluşturulmuştur.

Makine öğrenmesi algoritmasının performansını iyileştirmek için önce öznitelik seçimi ardından makine öğrenmesi algoritmaları uygulanmaktadır. Şekil 26 incelendiğinde; en sık SVM, KNN, C4.5, ANN, NB yöntemleri uygulanmadan önce öznitelik seçimi uygulanmıştır. Ancak topluluk öğrenme yöntemlerinden önce öznitelik seçiminin kullanımı henüz çok fazla çalışılmamıştır.

İncelenen makalelerde üçten fazla kez kullanılan anahtar kelimeler dikkate alınarak R programı ile oluşturulan kelime bulutu Şekil 27' de verilmiştir. Üç defadan daha az kullanılan anahtar kelimeler Tablo 3' te verilmiştir.



## 2.2. KULLANILAN VERİ SETLERİNİN LİTERATÜR İNCELEMESİ

Yapılan çalışmada kullanılan Wisconsin Diagnostic Breast Center'in (WDBC) mamogram veri seti ve GlaucomaM veri setinin literatür incelemesi Tablo 4 ve 5'te verilmiştir. Veri setlerine ilişkin detaylı bilgilere üçüncü bölümde değinilmiştir. Tablo 4 ve 5'te iyi performansa sahip makaleler verilmiştir. Tablolarda; makalenin kaynakçası, kullanılan yöntem ve en sık kullanılan performans ölçütlerinden Doğruluk (Accuracy), Duyarlılık (Sensitivity/Recall/True Positive Rate), Belirleyicilik/Özgüllük (Specificity/True Negative Rate) değerleri verilmiştir.

**Tablo 4:** WDBC Veri Setinin Literatür İncelemesi

Çalışma	Yöntem	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
Obaid ve diğerleri (2018)	SVM (Quadratic Kernel)	98.1		
Anagnostopoulos ve diğerleri (2006)	Diagnosis Neural Network		98.58	98.03
Aalaei ve diğerleri (2016)	ANN	97.3	98.6	96
Salama ve diğerleri (2012)	SMO	97.5		
Wang ve diğerleri (2018)	Weighted Area Under the Receiver Operating Characteristic Curve Ensemble (WAUCE)	97.68	98.63	99.95
Omondiagbe ve diğerleri (2019)	NN-LDA	98.82	98.41	99.07
Quinlan (1996)	C4.5	94.74		
Setiono (1996)	Neural Network Pruning	96.56		
Bennett ve Blue (1998)	SVM	97.20		
Setiono (2000)	ANN	97.97		
Bagui ve diğerleri (2003)	k-NN Rule	98.10		
Polat ve Güneş (2007)	LS-SVM	98.53		
Pach ve Abonyi (2006)	Fuzzy Decision Tree	95.27		
Akay (2009)	SVM	99.51		
Karabatak ve Ince (2009)	ANN	97.40		
Marcano-Cedeño ve diğerleri (2011)	Artificial Metaplasticity Multilayer Perceptron (AMMLP)	99.26		
Chen ve diğerleri (2011)	Rough Set Based Supporting Vector Machine Classifier (RS_SVM)	96.87		

Çalışma	Yöntem	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
Medjahed ve diğerleri (2013)	KNN	98.70		
Seera ve Lim (2014)	FMM-CART-RF	97.29		
Nahato ve diğerleri (2015)	RS-BPNN	98.60		
Kumar ve diğerleri (2017)	Naive Bayes + SMV + J48	97.13		
Latchoumi ve Parthiban (2017)	Weighted-Particle Swarm Optimization for Data Clustering in Sequence with Smooth Support Vector Machine	98.42		
Wolberg ve diğerleri (1994)		97		
Pena-Reyes ve Sipper (1999)	Combined Fuzzy-Genetic Approach	97.8		
Kıyan ve Yıldırım (2004)	General Regression Neural Network (GRNN)	98.8		
Rangarajan (2010)	Feature Selection (Mutual Correlation) + Feature Extraction (LPP)		88.60	
Sevli (2019)	LR	98.24	95.35	100
Liu ve diğerleri (2019)	Feature Selection (Information Gain Directed Simulated Annealing Genetic Algorithm Wrapper (IGSAGAW)); Classification (BP Neural Network)	97.5		
Ketabchi ve diğerleri (2019)	SVM	95.58 ± 4.42		
Nilashi ve diğerleri (2017)	EM-PCA-CART-Fuzzy Rule-Based	93.2		
Lavanya ve Rani (2011)	Feature Selection (SU); Classification (CART)	94.72		
Han ve diğerleri (2019)	CNN	98.24		
Ibrahim ve diğerleri (2019)	Feature Selection (Combines the Slap Swarm Algorithm with the Particle Swarm Optimization (SSAPSO)); Classification (KNN)	98.0		
Dora ve diğerleri (2017)	Gauss-Newton Representation Based Algorithm (GNRBA)		98.50	100
Sarafrazi ve Nezamabadi-Pour (2013)	Gravitational Search Algorithm (GSA) with Support Vector Machine (SVM)	96.38±1.46		
Maldonado ve diğerleri (2011)	Kernel-Penalized SVM (KP-SVM)	97.55 ± 0.9		
Sun ve Huang (2010)	Adaptive K-Nearest Neighbor Algorithm (AdaNN)	94		
Canuto ve Nascimento (2012)	SMO	97.54±1.49		

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 4 incelendiğinde, WDBC veri setinin genel olarak yüksek performans değerleri verdiği görülmektedir. GlaucomaM veri setinde ise Tablo 5'te görüldüğü gibi 90 değerinin üzerinde bir Accuracy değerine sahip çalışma bulunmamaktadır.

**Tablo 5:** GlaucomaM Veri Setinin Literatür İncelemesi

Çalışma	Yöntem	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
Hothorn ve Lausen (2003)	Double-Bagging	84,3		
Hothorn ve Lausen (2005)	Subbundling (SBund: both Combine Stabilized LDA (Slda), k-NN)	85		
Hothorn ve Lausen (2003)	Bagged Classification Trees (Bagged-CTREE)	85		
Tutz ve Koch (2016)	Weighted k-Nearest Neighbor	85		
Gul vd. (2018)	RF	90		
Khan ve diğerleri (2020)	Random Projection (Base = LDA)	90		
Gul ve diğerleri (2016)	Ensemble of Subset of Knn Classifiers (ESkNN)	89		
Adler ve diğerleri (2011)	Gradient Boosting (GB)	90		
Adler ve Lausen (2009)	RF	90		

Kaynak: Yazar tarafından oluşturulmuştur.

## ÜÇÜNCÜ BÖLÜM

### UYGULAMA

#### 3.1. VERİ SETİ

Yapılan çalışmada Wisconsin Diagnostic Breast Center'in (WDBC) mamogram veri seti ve GlaucomaM veri seti kullanılmıştır.

WDBC veri seti UCI makine öğrenimi veri deposundan alınmıştır. Bu veri seti, 30 tanımlayıcı özellik ve bir adet teşhis sınıfı (iyi huylu veya kötü huylu) olmak üzere toplam 31 özellik içeren 569 örnekten oluşmaktadır. 569 verinin sınıf dağılımı ise 357 iyi huylu (B - Benign), 212 kötü huylu (M - Malignant) örnek olmak üzere iki küme içermektedir. Yani veri seti dengeli dağılmaktadır. Veri setinde dengesiz dağıldığı durumlarda algoritmaların performansını iyileştirmek için farklı yaklaşımlar ortaya atılmıştır.

Veri setine ve daha detaylı bilgilere <https://tinyurl.com/424y5rw7> adresinden ulaşılabilir. Veri setinin özellikleri Tablo 6 'da verilmiştir. Yapılan çalışmada önerilen yaklaşımın ilki ardışık öznitelik seçimi, ikinci yaklaşım ise yığın algoritmasında farklı temel ve meta model denemesidir. Yığın algoritmasında denenecek model sayısı fazla olduğundan hesaplama işleminin daha kısa sürmesi için örnek hacmi küçük olan veri seti tercih edilmiştir. Ayrıca, ardışık öznitelik seçimi yapılacağından öznitelik sayısının da 20'den fazla olması istenmiştir. Bu veri setinde örnek sayısının çok fazla değil ve öznitelik sayısı da 20 'den fazla aynı zamanda sınıflar arasında dengeli dağılımın olduğu en popüler veri seti (Web Hits: 1809258) olduğundan tercih edilmiştir.

**Tablo 6:** Breast Cancer Wisconsin (Diagnostic) (WDBC) Veri Seti

<b>Data Set Characteristics:</b>	Multivariate	<b>Number of Instances:</b>	569	<b>Area:</b>	Life
<b>Attribute Characteristics:</b>	Real	<b>Number of Attributes:</b>	32	<b>Date Donated</b>	1995-11-01
<b>Associated Tasks:</b>	Classification	<b>Missing Values?</b>	No	<b>Number of Web Hits:</b>	1809258

Kaynak: UCI Machine Learning Repository, <https://tinyurl.com/424y5rw7>, 15.05.2023.

GlaucomaM veri seti R programında "TH.data" paketinde bulunmaktadır. Bu veri seti, 62 değişken ve bir teşhis sınıfı (glokom veya normal) olmak üzere toplam 63 öznitelik içeren 196 gözlemden oluşmaktadır. 196 verinin sınıf dağılımı ise 98 normal,

98 glokomlu şeklindedir. Tüm deęişkenler Heidelberg Retina Tomografisi tarafından alınan göz arka planının lazer tarama görüntüsünden elde edilmiştir. Gözlemler sırasıyla normal ve glokomlu gözlerden alınmıştır. Herhangi bir yanlılığı önlemek için her iki gruptaki gözlemler yaş ve cinsiyete göre eşleştirilmiştir.

Bu veri setinin de öznitelik sayısı 20 'den fazla ve aynı zamanda sınıflar arasında dengeli dağılım bulunmaktadır. Veri setine ilişkin data detaylı bilgi için Hothorn and Lausen (2003) adlı çalışma ya da "TH.data" paketi (<https://tinyurl.com/2e6bdt59>) incelenebilir.

### 3.2. KODLAMA

R, R Core Team ve R Foundation for Statistical Computing tarafından desteklenen istatistiksel hesaplama ve grafikler için bir programlama dili ve ücretsiz yazılım ortamıdır. Yeni Zelanda'daki Auckland Üniversitesi'nde İstatistikçiler Ross Ihaka ve Robert Gentleman tarafından geliştirilen S dili ve ortamına benzeyen bir GNU projesidir ve şu anda R Geliştirme Çekirdek Ekibi tarafından geliştirilmektedir. R, 1976 yılında John Chambers, Rick Becker ve Allan Wilks tarafından Bell laboratuvarlarından Bell Laboratories'de (eski adıyla AT&T, şimdiki adıyla Lucent Technologies) geliştirilen "S İstatistiksel Programlama Dili'nin" bir lehçesi olduğu söylenebilir. R, S' in bazı önemli farklılıklar vardır, ancak S için yazılan birçok kod R altında değiştirilmeden çalışabilmektedir. R veri ön işleme, çok çeşitli istatistiksel (doğrusal ve doğrusal olmayan modelleme, klasik istatistiksel testler, zaman serisi analizi, sınıflandırma, kümeleme, ...) analiz ve grafiksel teknikler sağlayan en yaygın programlama dillerinden biridir. R programını, paketlerini indirmek ve bilgi almak için <https://cran.r-project.org/> adresi kullanılabilir. Mevcut paketler <https://tinyurl.com/44hwveh9> adresinden incelenebilir.

IEEE, her yıl en popüler programlama dillerinin bir listesini yayınlamaktadır. R, 2019'da 5. iken 2020'de 6. sırada, 2021 yılında ise 7.sırada yer almaktadır. R gibi alana özgü bir dilin genel amaçlı bir dillerden daha popüler olması önemlidir. Bu, yalnızca bir programlama dili olarak R'ye artan ilgiyi değil, aynı zamanda R'nin yaygın olarak kullanıldığı Veri Bilimi ve Makine Öğrenimi gibi alanlara da artan ilgiyi göstermektedir. Aynı zamanda Mart 2022 itibarıyla R, programlama dili popülerliğinin bir ölçüsü olan TIOBE endeksinde de 11. sıradayken; dil Ağustos 2020'de 8. sırada yer almaktadır (<https://tinyurl.com/mry5r8dh>).

R dili devamlı gelişim içindedir, makine öğrenmesi algoritmaları da devamlı gelişmekte ve yeni paketler ortaya çıkmaktadır. R, 10.0000'den fazla paket içeren bir depo olan CRAN'a (Comprehensive R Archive Network -Kapsamlı R Arşiv Ağı) sahiptir. R'nin en önemli özelliklerinden biri, geniş bir kütüphane mevcudiyetine sahip olması aynı zamanda kullanımı kolay arayüzü sayesinde de son yıllarda kullanımı artmaktadır.

Bu çalışmada R 3.6.1 tabanlı çalışan RStudio 1.4 versiyonu ile uyumlu paketler arasından seçim yapılarak algoritmalar uygulanmıştır. <https://tinyurl.com/mw4fja9u> adresinden farklı algoritmaları inceleyebilirsiniz. R programında; FSelector, FSelectorRcpp, FSinR, caret, Boruta öznelik seçimine ilişkin paketler kullanılarak öznelik seçimi gerçekleştirilmiş. R programında; caret, naivebayes, kernlab, MASS, RSNNs, emtree, rpart, C50, plyr, Rweka, Earth, randomForest, gbm, xgboost, ipred, e1071, fastAdaboost, mboost, caTools, rotationForest, caretEnsemble paketleri kullanılarak makine öğrenmesi algoritmaları uygulanmıştır. Ayrıca, farklı paketleri kullanarak da algoritmaları uygulamak mümkündür.

### 3.3. ÖNERİLEN YAKLAŞIM

Yapılan çalışmada sınıflandırma probleminin çözümünde kullanılan makine öğrenmesi algoritmalarının performansını iyileştirmek amaçlanmıştır. Bunu gerçekleştirmek için iki farklı yaklaşım denenmiştir. İlk yaklaşım ardışık öznelik seçimidir. Algoritmaların performansı etkileyen en önemli unsur elimizdeki veri setindeki özneliklerin yeterli sayıda ve gerekli olmasıdır. Bunun için sınıflandırma işlemini gerçekleştirmeden önce öznelik seçimi yapılmıştır. Öznelik seçimi olarak da filtreleme ve sarmal yöntemleri birleştiren hibrit bir yaklaşım kullanılmıştır. Bu hibrit yaklaşımla, sadece filtreleme yönteminin kullanıldığı durumdaki sınıflandırma doğruluğunu iyileştirmekle kalmıyor, aynı zamanda sadece sarmal yöntemler kullanıldığı durumdaki işlem süresi de azaltılabilmektedir. Algoritmanın performansını geliştirmek için diğer bir denenmiş yaklaşım ise, yığın algoritmasında temel ve meta model olarak farklı algoritmalar denenerek daha iyi performansla sahip model bileşimi bulunmaya çalışılmıştır. Yığın algoritmasında kullanılan temel ve meta modellere göre algoritmanın performansı değişmektedir. Temel ve meta model olarak en iyi performansı verebilecek algoritmalara ilişkin genel geçer bir kabul bulunmamaktadır. Farklı modeller denenerek en iyi performansla sahip model bulunmaya çalışılmıştır.

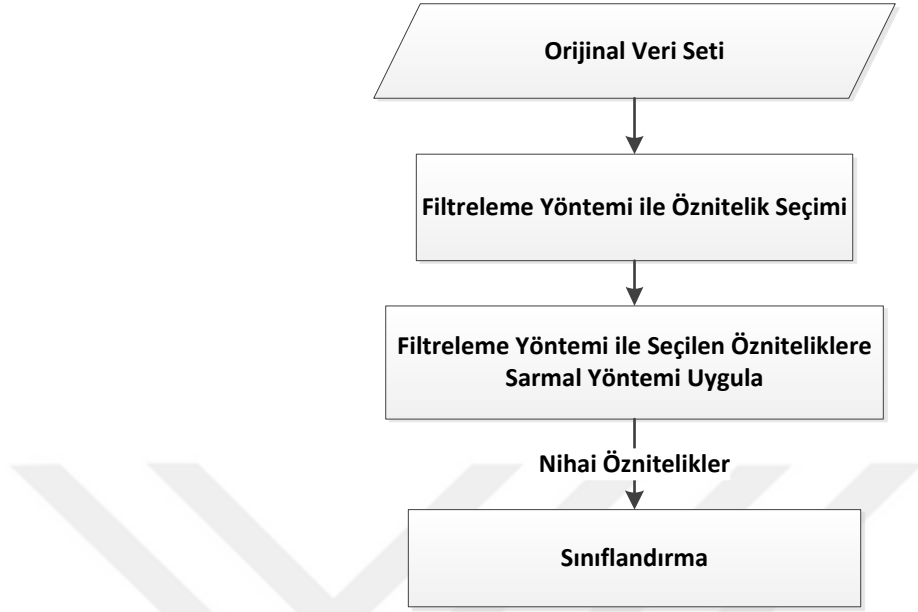
### 3.3.1. Filtreleme ve Sarmal Yöntemleri Birleştiren Hibrit Öznitelik Seçimi

Filtreleme yönteminde öznitelik seçimi, herhangi bir sınıflandırma algoritması kullanılmadan, uzaklık, bilgi, bağımlılık ve tutarlılık ölçümleri gibi istatistiksel kriterlere dayanan fonksiyonlar kullanılarak yapılmaktadır. Öznitelik seçme ve sınıflandırma işlemleri birbirinden ayrı ve bağımsız olarak gerçekleştirilmektedir. Sarmal yöntemde ise makine öğrenme algoritmaları kullanır ve nitelik seçme ölçütü sınıflandırıcının doğruluk oranıdır. Filtreleme yöntemlerinin en büyük avantajı hesaplama zamanı açısından sarmal yöntemlere göre oldukça hızlı olmasıdır. Bu yöntemin en büyük dezavantajı ise seçim kriterinin doğrudan model etkinliğiyle ilgili olmamasıdır, bu yüzden sınıflandırma performansı zayıftır. Sarmal yöntemlerin en büyük avantajı doğrudan modeli iyileştiren özellikler kümesinin seçimine imkân sağladığından çoğunlukla yüksek performans göstermektedir. Ancak bu yöntemler çoğunlukla sezgisel algoritmalara dayalı oldukları için kullanılan algoritmanın girdi parametrelerinin ayarlanması (parameter tuning) önemlidir.

Yapılan çalışmada, filtreleme ve sarmal olmak üzere iki öznitelik seçim yönteminin avantajlarını kullanarak birleştiren Şekil 28'de akış diyagramı verilmiş olan hibrit özellik seçim yöntemi uygulanmıştır. Öznitelikler ilk olarak hesaplama açısından daha hızlı olan filtreleme yöntemleri aracılığıyla orijinal öznitelik kümesinden seçilir. Filtreleme yöntemleri ile seçilen özniteliklere sarmal yöntemler uygulanarak daha da iyileştirilmektedir. Sarmal yöntemlerle nihai öznitelikler elde edilerek sınıflandırma işlemi yapılır.

Yapılan çalışmada, filtreleme adımında Ki-Kare, Bilgi Kazancı, Kazanç Oranı, Simetrik Belirsizlik Katsayısı, Korelasyon Tabanlı Öznitelik Seçimi ve RELIEF yöntemleri kullanılırken; sarmal yöntem olarak Özyinelemeli Öznitelik Eliminasyonu, Genetik Algoritma, Tavlama Benzetimi ve BORUTA kullanılmıştır. Kullanılan öznitelik seçim yöntemleri Tablo 7'de verilmiştir. Bölüm 3.6'daki sonuçlar incelendiğinde, ardışık öznitelik seçiminin algoritmaların hem hesaplama zamanı hem de diğer performans ölçütleri açısından sonuçları iyileştirdiği görülmektedir.

**Şekil 28:** Önerilen Öznitelik Seçim Yaklaşımının Akış Diyagramı



Kaynak: Yazar tarafından oluşturulmuştur.

**Tablo 7:** Kullanılan Ardışık Öznitelik Seçim Yöntemleri

Öznitelik Seçim Yöntemleri		
IG+RFE	CFS+RFE	RELİEF+RFE
IG+GA	CFS +GA	RELİEF +GA
IG+SA	CFS +SA	RELİEF +SA
IG+BORUTA	CFS+BORUTA	RELİEF +BORUTA
GR+RFE	Kİ-KARE+RFE	SU+RFE
GR+GA	Kİ-KARE +GA	SU+GA
GR+SA	Kİ-KARE +SA	SU+SA
GR+BORUTA	Kİ-KARE +BORUTA	SU+BORUTA

Kaynak: Yazar tarafından oluşturulmuştur.

### 3.3.2. Yiğın Algoritması Farklı Model Denemeleri

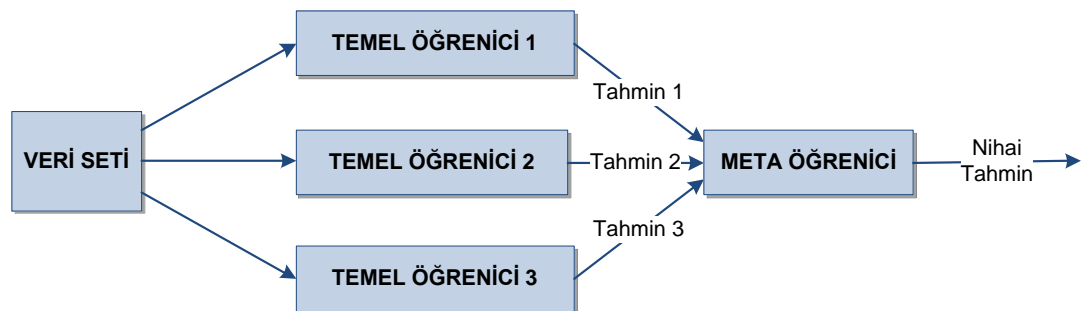
Yiğın algoritmasının iki seviyeli bir sınıflandırma algoritmasıdır. Birinci seviyede ikiden fazla temel öğrenici eğitilir, daha sonra birinci seviye öğrenicilerin tahminleri ikinci seviye olan meta seviye sınıflandırıcı tarafından birleştirilmektedir. Yiğılmış genelleme fikrini Wolpert (1992) ortaya atmıştır. Ting ve Witten (1997) farklı 2. düzey modelleri (meta öğrenici) deneysel olarak karşılaştırmış. Temel model olarak

C4.5, NB, IB1; meta model olarak C4.5, NB, IB1, MLR kullanmış ve basit doğrusal modelin (MLR) en iyi performansı gösterdiğini bulmuşlardır.

Yapılan çalışmada, Şekil 29'da görüldüğü gibi birinci seviyede üç temel öğrenici kullanılmıştır. Temel öğrenici olarak; k-Nearest Neighbors (knn), Naive Bayes (naive\_bayes), CART (rpart), Bagged CART (treebag), J48, C5.0, eXtreme Gradient Boosting (xgbTree), Linear Discriminant Analysis (lda), Multi-Layer Perceptron (mlp), Multivariate Adaptive Regression Spline (earth), Random Forest (rf), Rotation Forest (rotationForest), gbm, Support Vector Machines with Linear Kernel (svmLinear), Support Vector Machines with Polynomial Kernel (svmPoly), Support Vector Machines with Radial Basis Function Kernel (svmRadial), Tree Models from Genetic Algorithms (evtree), Generalized Linear Model (glm) algoritmaları kullanarak denebilecek olası tüm modeller (816 model) Ek1' de verilmiştir. Meta öğrenici olarak; adaboost, treebag, glmboost, LogitBoost, J48, C5.0, rpart, xgbTree, knn, lda, mlp, earth, naive\_bayes, rf, rotationForest, svmLinear, svmPoly, svmRadial, glm algoritmaları kullanılmıştır. Olası tüm temel modellerin her biri (816 model), tüm meta modellerin her biri (19 model) denemiştir toplamda 15504 (816\*19) yığın algoritması denenmiştir. Literatürde bu kadar kapsamlı bir model denemesi bulunmamaktadır yani bu konuda önemli eksiklik bulunmaktadır.

Yığın algoritmasında birinci ve ikinci seviyede herhangi bir öğrenme algoritması kullanılabilir. Hangi modeller kullanılarak daha iyi performans elde edileceğine ilişkin genel geçer bir kabul olmadığından yani deneme yanılma ile kullanılacak modeller belirlendiğinden diğer topluluk öğrenmesi yöntemlerinden daha az kullanılmaktadır. Literatürde bu kadar kapsamlı bir model denemesi bulunmamaktadır yani bu konuda önemli eksiklik bulunmaktadır. Bölüm 3.5 ve Bölüm 3.6'daki sonuçlar incelendiğinde Yığın algoritmasının en iyi performansı verdiği ve farklı model denemelerinin performansı etkilediği görülmektedir.

**Şekil 29:** Kullanılan Yığın Algoritması



Kaynak: Yazar tarafından oluşturulmuştur.

### 3.4. TEMEL ALGORİTMA SONUÇLARI

Bu bölümde, k-Nearest Neighbors (KNN), Naive Bayes (NB), CART (rpart), Bagged CART (TREEBAG), J48, C5.0, eXtreme Gradient Boosting (XGBTREE), Linear Discriminant Analysis (LDA), Multi-Layer Perceptron (MLP), Multivariate Adaptive Regression Spline (EARTH), Random Forest (RF), Rotation Forest, Gradient Boosting Machine (GBM), Support Vector Machines with Linear Kernel (SVMLINEAR), Support Vector Machines with Polynomial Kernel (SVMPoly), Support Vector Machines with Radial Basis Function Kernel (SVMRADIAL), Tree Models from Genetic Algorithms (EVTREE) ve Generalized Linear Model (GLM) algoritmalarını ilişkin sonuçlar verilmiştir.

#### 3.4.1. GlaucomaM Veri Setinin Temel Algoritma Sonuçları

GlaucomaM veri seti, 62 değişken ve bir teşhis sınıfı (glokom veya normal) olmak üzere toplam 63 öznitelik içeren 196 gözlemden oluşmaktadır. R programında veri setinin pozitif sınıfı glokom' dur. Veri setine ilişkin karmaşıklık matrisi Tablo 8'de verilmiştir.

**Tablo 8:** GlaucomaM Veri Setinin Karmaşıklık Matrisi

		Tahmin Edilen Sınıf	
		Glokom	Normal
Gerçek Sınıf	Glokom	True Glokom	False Normal
	Normal	False Glokom	True Normal

Kaynak: Yazar tarafından oluşturulmuştur.

Yapılan çalışmada sıklıkla kullanılan doğruluk, duyarlılık ve belirleyicilik ölçütleri kullanılarak algoritmaların performansları kıyaslanmıştır. Diğer performans ölçütlerine ilişkin sonuçlara <https://tinyurl.com/4rzmjyc7> linkinden erişilebilir. Doğruluk, doğru sınıflandırılmış örnek sayısının, toplam örnek sayısına oranıdır. Duyarlılık ölçütü, modelin pozitif sınıf etiketini tahmin etmedeki etkililiğini göstermektedir. Yani glokomlu hastaların doğru tespit edebilme oranını gösterir. Bu ölçüt, tıp alanında çok kıymetlidir. Çünkü hasta vakaları doğru tespit edebilmek çok önemlidir. Belirleyicilik ölçütü ise modelin negatif sınıf etiketini tahmin etmedeki etkililiğini göstermektedir. Yani normal (sağlıklı) vakaları doğru tespit edebilme oranını

göster. Bu ölçüt de tıp alanında önemlidir çünkü sağlıklı vakaların da tespit edilmesi gerekmektedir.

Veri setini eğitim ve test olarak bölmek için Holdout yöntemi kullanılmıştır. Veri setinin %80'i eğitim, %20 si test kümesi olarak kullanılmıştır. Kodlara <https://tinyurl.com/2djuapkb> linkinden erişilebilir. GlaucomaM veri seti için temel algoritma sonuçları Tablo 9'da verilmiştir.

**Tablo 9:** GlaucomaM Veri Seti için Makine Öğrenmesi Algoritmalarının Sonuçları

ALGORİTMA	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
GLM	75	80,95	68,42
KNN	77,5	85,71	68,42
NB	72,5	80,95	63,15
SVMLİNEAR	85	85,71	84,21
SVMPOLY	<b>90</b>	90,47	89,47
SVMRADIAL	87,5	90,47	84,21
LDA	77,5	80,95	73,68
MLP	87,5	90,47	84,21
EVTREE	80	66,67	<b>94,73</b>
RPART	80	90,47	68,42
C5.0	80	80,95	78,94
J48	80	85,71	73,68
EARTH	82,5	80,95	84,21
RF	85	90,48	78,95
GBM	<b>90</b>	<b>95,24</b>	84,21
XGBTREE	85	85,71	84,21
TREEBAG	82,5	85,71	78,95
ADABOOST	<b>90</b>	90,48	89,47
GLMBOOST	77,5	80,95	73,68
LOGİTBOOST	85	85,71	84,21
ROTATION FOREST	82,5	85,71	78,95

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 9 incelendiğinde; kullanılan algoritma ve performans ölçütlerinin değerleri (Doğruluk, Duyarlılık ve Belirleyicilik) verilmiştir. Amaç Glokomlu hastaları doğru tespit edebilmek ise, Duyarlılık değerine bakılmalıdır. Buna göre en iyi sonucu GBM algoritması vermiştir. Normal (sağlıklı) vakaları doğru tespit edebilmek istediğimizde Belirleyicilik değerine bakılmalıdır. Buna göre, EVTREE algoritması en iyi sonucu vermiştir. Doğruluk değerine göre Polinomial çekirdek fonksiyonun kullanıldığı SVM (SVMPoly) (28,78 s.), GBM (26,3 s.) ve Adaboost (14 m.) algoritmaları en iyi sonucu vermiştir. Üç algortmada aynı doğruluk değerine sahip olduğundan minimum hesaplama zamanına sahip algoritma GBM en iyi sonucu vermektedir.

### 3.4.2. WDBC Veri Setinin Temel Algoritma Sonuçları

WDBC veri seti, 30 tanımlayıcı özellik ve bir adet teşhis sınıfı (iyi huylu veya kötü huylu) olmak üzere toplam 31 özellik içeren 569 örnekten oluşmaktadır. R programında veri setinin pozitif sınıfı iyi huylu (B - Benign)' dur. Veri setine ilişkin karmaşıklık matrisi Tablo 10'da verilmiştir.

**Tablo 10:** WDBC Veri Setinin Karmaşıklık Matrisi

		Tahmin Edilen Sınıf	
		Benign	Malignant
Gerçek Sınıf	Benign	True Benign	False Malignant
	Malignant	False Benign	True Malignant

Kaynak: Yazar tarafından oluşturulmuştur.

Yapılan çalışmada sıklıkla kullanılan doğruluk, duyarlılık ve belirleyicilik ölçütleri kullanılarak algoritmaların performansları kıyaslanmıştır. Diğer performans ölçütlerine ilişkin sonuçlara <https://tinyurl.com/yck4res5> linkinden erişilebilir. Doğruluk, doğru sınıflandırılmış örnek sayısının, toplam örnek sayısına oranıdır. Duyarlılık ölçütü, modelin pozitif sınıf etiketini tahmin etmedeki etkililiğini göstermektedir. Yani iyi huylu tümörü (sağlıklı) olan hastaların doğru tespit edebilme oranını gösterir. Bu ölçüt tıp alanında önemlidir çünkü sağlıklı vakaların da tespit edilmesi gerekmektedir. Belirleyicilik ölçütü ise modelin negatif sınıf etiketini tahmin etmedeki etkililiğini göstermektedir. Yani kötü huylu tümöre sahip vakaları doğru tespit edebilme oranını göster. Bu ölçüt, tıp alanında çok kıymetlidir. Çünkü hasta vakaları doğru teşhis edebilmek çok önemlidir. Genel olarak, kötü huylu bir tümörü doğru bir şekilde tanımlamak, iyi huylu bir tümörü yanlış tanımlamaktan daha önemlidir.

**Tablo 11:** WDBC Veri Seti için Makine Öğrenmesi Algoritmalarının Sonuçları

ALGORİTMA	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
GLM	93,86	97,18	88,37
KNN	92,98	100	81,39
NB	92,98	95,77	88,37
SVMLİNEAR	<b>97,37</b>	<b>100</b>	<b>93,02</b>
SVMPOLY	96,49	<b>100</b>	90,70
SVMRADIAL	<b>97,37</b>	<b>100</b>	<b>93,02</b>
LDA	94,74	<b>100</b>	86,04
MLP	62,28	<b>100</b>	0
EVTREE	94,74	98,59	88,37
RPART	91,23	95,77	83,72
C5.0	95,61	100	88,37
J48	91,23	94,37	86,05
EARTH	95,61	98,59	90,70
RF	96,49	<b>100</b>	90,70
GBM	95,61	98,59	90,70
XGBTREE	95,61	98,59	90,70
TREEBAG	95,61	98,59	90,70
ADABOOST	96,49	<b>100</b>	90,70
GLMBOOST	94,74	98,59	88,37
LOGİTBOOST	94,74	98,59	88,37
ROTATION FOREST	96,49	<b>100</b>	90,70

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 11 incelendiğinde; kullanılan algoritma ve performans ölçütlerinin değerleri (Doğruluk, Duyarlılık ve Belirleyicilik) verilmiştir. Amaç iyi huylu tümörü (sağlıklı) olan hastaları doğru tespit edebilmek ise, Duyarlılık değerine bakılmalıdır. Duyarlılık değerine göre en iyi sonucu SVMLİNEAR (2.51 s.), SVMPOLY (38.9 s.), SVMRADIAL (5.9 s.), LDA (2.7 s.), MLP (48.2 s.), RF (54.8 s.), ADABOOST (14.4 m.) ve ROTATION FOREST (2.2 m.) algoritmaları vermiştir. Bu algoritmalar minimum hesaplama zamanına SVMLİNEAR algoritması sahiptir. Kötü huylu tümöre sahip hastaları doğru tespit edebilmek istediğimizde Belirleyicilik değerine bakılmalıdır. Belirleyicilik değerine göre, SVMLİNEAR (2.51 s.) ve SVMRADIAL (5.9 s.) algoritmaları en iyi sonucu vermiştir. Bu algoritmalar minimum hesaplama zamanına SVMLİNEAR algoritması sahiptir. Doğruluk değerine göre doğrusal çekirdek fonksiyonun kullanıldığı SVM (SVMLİNEAR) (2.51 s.) ve radyal çekirdek fonksiyonun kullanıldığı SVM (SVMRADIAL) (5.9 s.) algoritmaları en iyi sonucu vermiştir. Bu algoritmalar aynı doğruluk değerine sahip olduğundan minimum hesaplama zamanına sahip algoritma GBM en iyi sonucu vermektedir. Üç ölçütte dikkate

alındığında en iyi sonucu veren algoritma doğrusal çekirdek fonksiyonun kullanıldığı SVM (SVMLİNEAR) algoritmasıdır.

### 3.5. YIĞIN ALGORİTMASI SONUÇLARI

Yığın algoritmasının iki seviyeli bir sınıflandırma yaklaşımıdır. Birinci seviyede ikiden fazla temel öğrenici eğitilir, daha sonra birinci seviye öğrencilerin tahminleri ikinci seviye olan meta seviye sınıflandırıcı tarafından birleştirilmektedir. Yığın algoritmasında birinci ve ikinci seviyede herhangi bir öğrenme algoritması kullanılabilir. Hangi modeller kullanılarak daha iyi performans elde edileceğine ilişkin genel geçer bir kabul bulunmamaktadır yani deneme yanılma ile kullanılacak modeller belirlenmektedir. Yapılan çalışmada, Bölüm 3.3.2 'de de değindiğimiz farklı temel model ve meta modeller denenmiştir. Denen 15504 modeli ilişkin sonuçların tamamına <https://tinyurl.com/2p8ts6sb> linkinden erişebilirsiniz. Burada sadece belirli değerler üzerinde performans gösteren modeller verilmiştir. Yığın algoritması koduna <https://tinyurl.com/494decsc> linkinden erişebilirsiniz.

#### 3.5.1. GlaucomaM Veri Setinin Yığın Algoritması Sonuçları

Tablo 12'de GlaucomaM veri seti için %95 ve üzeri doğruluk değerine sahip modeller verilmiştir. Tablo 12' de Temel Model (Ek 1'de temel model olarak kullanılan algoritmalar verilmiştir), meta model, performans değerlendirme ölçütleri (Accuracy, Duyarlılık ve Belirleyicilik) ve hesaplama zamanı verilmiştir.

**Tablo 12:** GlaucomaM Veri Seti için %95 Üzeri Doğruluk Veren Yığın Algoritması Sonuçları

Temel Model	Meta Model	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)	Zaman
Model 56	knn	95	100	89,47	1.178 min
Model 91	knn	95	95,24	94,74	9.898 sec
Model 98	treebag	95	100	89,47	43.238 sec
	C5.0	95	100	89,47	43.869 sec
	rf	95	100	89,47	48.297 sec
Model 99	C5.0	95	95,24	94,74	3.602 min
	earth	95	95,24	94,74	3.473 min
Model 100	adaboost	95	95,24	94,74	5.779 min
Model 134	rf	95	100	89,47	2.895 min
Model 135	xgbTree	95	95,24	94,74	6.314 min
Model 213	knn	95	95,24	94,74	38.245 sec
Model 214	J48	95	95,24	94,74	2.289 min
	C5.0	95	95,24	94,74	1.979 min

Temel Model	Meta Model	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)	Zaman
	rpart	95	95,24	94,74	2.096 min
Model 216	xgbTree	95	95,24	94,74	6.482 min
Model 219	svmPoly	95	95,24	94,74	2.559 min
Model 231	LogitBoost	95	95,24	94,74	28.672 sec
Model 259	knn	95	95,24	94,74	7.190 min
Model 291	knn	95	95,24	94,74	28.323 sec
Model 409	LogitBoost	95	95,24	94,74	51.098 sec
	rpart	95	95,24	94,74	43.105 sec
Model 476	knn	95	95,24	94,74	42.888 sec
Model 484	svmRadial	95	90,48	100	2.205 min
Model 511	mlp	95	95,24	94,74	2.284 min
	svmLinear	95	95,24	94,74	2.027 min
Model 512	LogitBoost	95	100	89,47	2.663 min
Model 570	adaboost	95	90,48	100	4.233 min
Model 575	adaboost	97,5	95,24	100	3.231 min
Model 659	adaboost	95	95,24	94,74	4.741 min
Model 660	rotationForest	95	95,24	94,74	48.141 sec
Model 727	xgbTree	95	100	89,47	11.983 min
Model 736	xgbTree	95	100	89,47	14.936 min
Model 785	treebag	95	95,24	94,74	7.992 min
	rf	95	95,24	94,74	8.017 min

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 12 incelendiğinde, denenmiş yığın algoritmalarından temel model olarak: Model 575 (C5.0, rf, glm); meta model olarak: adaboost algoritması kullanıldığında en iyi sonucu vermektedir. Tablo 9 ile 12 kıyaslandığında; temel makine öğrenmesi algoritmalarında en yüksek doğruluk değeri %90 iken; yığın algoritması ile %97 doğruluk değeri elde edilmiştir. Ayrıca, daha önce de değinildiği gibi literatürde meta model olarak basit doğrusal modeller veya yapraklarında doğrusal modeller bulunan ağaçların genellikle iyi çalıştığını öne sürmüştür ve genel olarak bu modeller meta modeller olarak kullanılmıştır. Bu çalışmada meta model olarak topluluk öğrenmesi yöntemleri de denenerek performansı incelenmiş ve bu veri setine ilişkin sonuçlar da incelendiğinde topluluk öğrenmesi metotlarının iyi sonucu verdiği görülmüştür. Meta model olarak topluluk öğrenmesi yöntemleri de kullanılmalıdır.

### 3.5.2. WDBC Veri Setinin Yığın Algoritması Sonuçları

Tablo 13'de WDBC veri seti için %98 ve üzeri doğruluk değerine sahip modeller verilmiştir. Bu veri setinde %98 değerinin üzerindeki modellerin verilmesinin nedeni veri setinin çoğunlukla 95 ve üzeri doğruluk değerine sahip yani yüksek

performans değerine sahip olduğundan en iyi sonuçları daha da özetleyebilmek adına 95 yerine 98 doğruluk değerine sahip modeller verilmiştir. Tablo 13' te Temel Model (Ek 1'de temel model olarak kullanılan algoritmalar verilmiştir), meta model, performans değerlendirme ölçütleri (Accuracy, Duyarlılık ve Belirleyicilik) ve hesaplama zamanı verilmiştir.

**Tablo 13:** WDBC Veri Seti için %98 Üzeri Doğruluk Veren Yığın Algoritması Sonuçları

Temel Model	Meta Model	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)	Zaman (sn)
Model 29	J48	98,25	100	95,35	56.452 sec
	<b>rpart</b>	<b>98,25</b>	<b>100</b>	<b>95,35</b>	<b>20.309 sec</b>
	rotationForest	98,25	100	95,35	37.441 sec
Model 83	earth	98,25	100	95,35	37.708 sec
Model 98	J48	98,25	100	95,35	1.317 min
	naive_bayes	98,25	100	95,35	58.941 sec
Model 106	earth	98,25	100	95,35	47.610 sec
Model 134	lda	98,25	100	95,35	8.827 min
Model 244	LogitBoost	98,25	100	95,35	41.984 sec
Model 254	J48	98,25	100	95,35	9.583 min
	rpart	98,25	100	95,35	9.024 min
Model 268	J48	98,25	100	95,35	55.197 sec
	C5.0	98,25	100	95,35	54.735 sec
Model 323	J48	98,25	100	95,35	1.337 min
	C5.0	98,25	100	95,35	1.256 min
	rpart	98,25	100	95,35	58.087 sec
	rotationForest	98,25	100	95,35	1.268 min
Model 344	glmboost	98,25	100	95,35	2.685 min
	J48	98,25	100	95,35	3.108 min
	lda	98,25	100	95,35	1.444 min
	rotationForest	98,25	100	95,35	3.300 min
Model 359	glmboost	98,25	100	95,35	9.425 min
	LogitBoost	98,25	100	95,35	9.422 min
	J48	98,25	100	95,35	10.399 min
	C5.0	98,25	100	95,35	9.987 min
	lda	98,25	100	95,35	9.762 min
	rotationForest	98,25	100	95,35	9.977 min
Model 360	rpart	98,25	100	95,35	24.284 sec
Model 384	rpart	98,25	100	95,35	1.462 min
Model 398	LogitBoost	98,25	100	95,35	55.362 sec
Model 429	LogitBoost	98,25	100	95,35	1.239 min
Model 435	rpart	98,25	100	95,35	2.715 min
Model 450	C5.0	98,25	100	95,35	6.678 min
Model 544	rf	98,25	100	95,35	1.609 min
Model 579	J48	98,25	100	95,35	5.142 min
	C5.0	98,25	100	95,35	3.256 min
Model 594	J48	98,25	100	95,35	10.463 min
	C5.0	98,25	100	95,35	10.176 min
	rotationForest	98,25	100	95,35	8.083 min
Model 595	rpart	98,25	100	95,35	41.409 sec
Model 758	rpart	98,25	100	95,35	5.447 min
Model 780	rpart	98,25	100	95,35	39.888 sec
Model 795	rpart	98,25	100	95,35	1.792 min

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 13 incelendiğinde, %98 üzeri doğruluk veren denenmiş yığın algoritmaları aynı Duyarlılık ve Belirleyicilik değerlerine sahip olduğundan minimum hesaplama zamanına sahip algoritma seçilmelidir. Buna göre, temel model olarak: Model 29 (Knn, rpart, svmRadial); meta model olarak: rpart algoritması kullanıldığında en iyi sonucu vermektedir. Tablo 11 ile 13 kıyaslandığında; temel makine öğrenmesi algoritmalarında en yüksek doğruluk değeri %97 iken; yığın algoritması ile %98 doğruluk değeri elde edilmiştir.

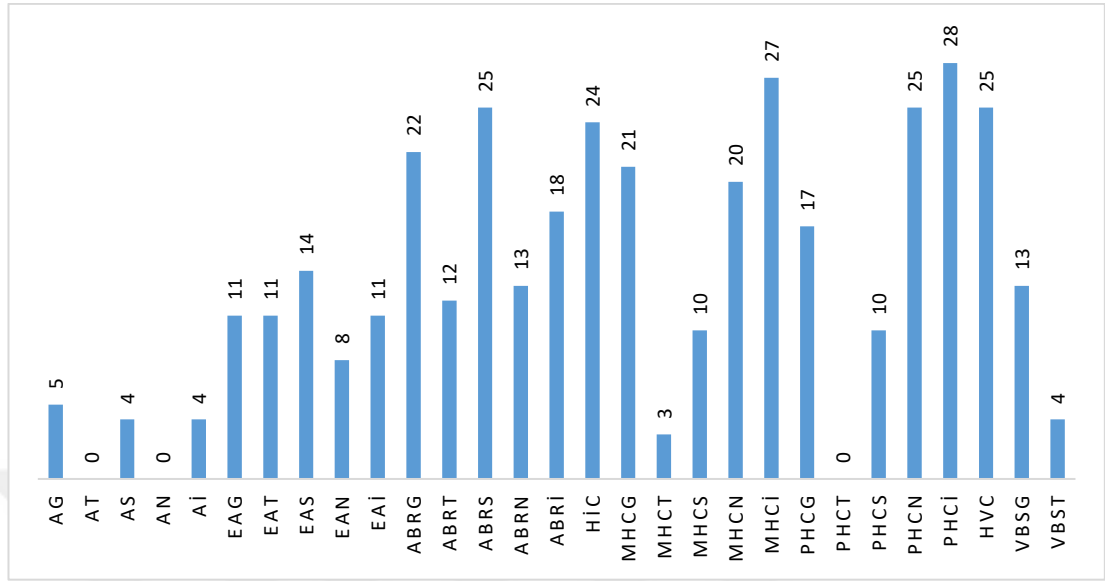
### **3.6. ARDIŞIK ÖZNETELİK SEÇİMİ SONUÇLARI**

Yapılan çalışmada, filtreleme ve sarmal olmak üzere iki öznetelik seçim yöntemlerinin avantajlarını kullanarak birleştiren hibrit özellik seçim yöntemi uygulanmıştır. Öznetelikler ilk olarak filtreleme yöntemleri aracılığıyla orijinal öznetelik kümesinden seçilir. Filtreleme yöntemleri ile seçilen özneteliklere sarmal yöntemler uygulanarak daha da iyileştirilmektedir. Sarmal yöntemlerle nihai öznetelikler elde edilerek sınıflandırma işlemi yapılmaktadır. Bölüm 3.3.1 'de denenen ardışık öznetelik seçim yöntemlerine değinilmiştir. Denenmiş olan hibrit yaklaşıma ilişkin sonuçlar bu kısımda verilmiştir.

#### **3.6.1. GlaucomaM Veri Setinin Ardışık Öznetelik Seçimi Sonuçları**

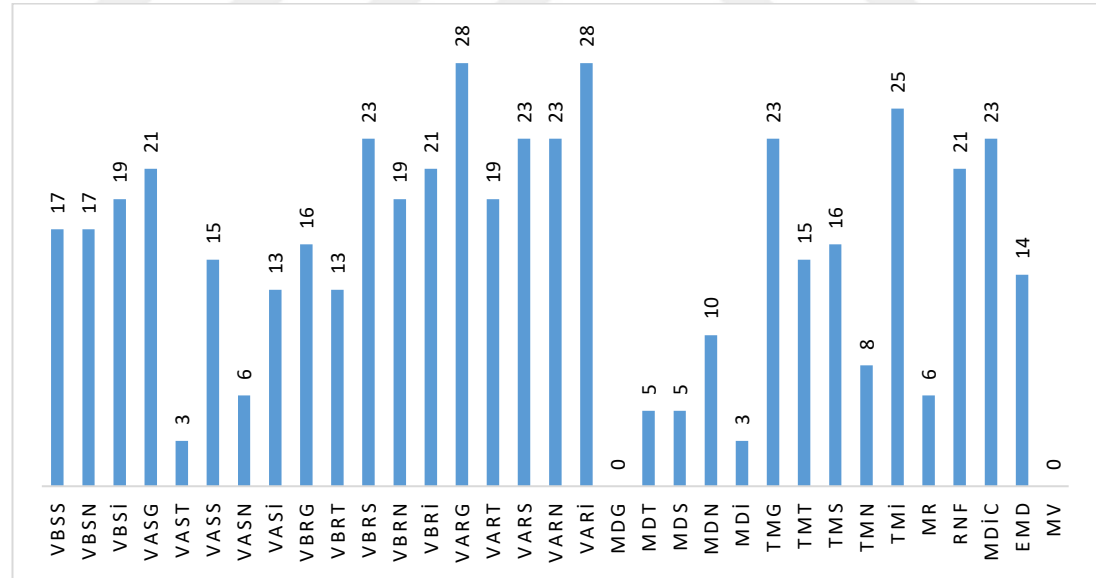
GlaucomaM veri setinde 62 değişken ve bir teşhis sınıfı (glokom veya normal) olmak üzere toplam 63 öznetelik bulunmaktadır. Her bir öznetelik seçim yöntemi tarafından seçilen değişkenler dikkate alınarak seçilme sıklıkları elde edilmiştir. Öznetelik seçimi sonrası değişkenlerin seçilme sıklıkları Şekil 30 ve 31 de verilmiştir. Şekiller incelendiğinde; varg, vari, phci, mhci, tmi, phcn, abrs, hvc, tmi, vbrs, vars, varn, tmg, mdic, abrg, mhcg, vbri ve rnf öznetelikleri yirmi ve üzeri defa seçilmiştir yani en önemli değişkenlerdir. Mdg, mv, phct, an ve at öznetelikleri ise hiçbir öznetelik seçim yaklaşımı tarafından seçilmediğinden en önemsiz değişkenlerdir.

**Şekil 30:** GlaucomaM Veri Seti için Değişkenlerin Seçilme Sayısı



Kaynak: Yazar tarafından oluşturulmuştur.

**Şekil 31:** GlaucomaM Veri Seti için Değişkenlerin Seçilme Sayısı



Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 14'te ardışık öznelik seçimi sonrası seçilen değişkenleri sayıları ve hesaplama zamanları verilmiştir. Değişken sayımız ne kadar azsa, algoritmaların karmaşıklığı azalarak anlaşılabilirliği artmaktadır aynı zamanda öğrenme süresi de

kısalmakta algoritmaların hızı artmaktadır. Ayrıca önemsiz, ilgisiz, gürültülü ve yanıltıcı değişkenlerin çıkarılması algoritmaların performansını önemli ölçüde artırmaktadır. Tablo 14 incelendiğinde, en az değişkene CFS ardından GA uygulanmış hibrit öznitelik yaklaşımı; en fazla değişkene ise RELİEF ardından RFE uygulanmış hibrit öznitelik yaklaşımının sahip olduğu görülmektedir. Ayrıca, hesaplama zamanı açısından incelediğimizde en az süreye CFS ardından BORUTA uygulanmış hibrit öznitelik yaklaşımı; en fazla süreye ise SU ardından GA uygulanmış hibrit öznitelik yaklaşımının sahip olduğu görülmektedir

**Tablo 14:** GlaucomaM Veri Seti için Ardışık Öznitelik Seçimi ile Seçilen Değişken Sayıları ve Hesaplama Zamanı

ÖZİNTELİK SEÇİM YÖNTEMİ	DEĞİŞKEN SAYISI	Zaman	ÖZİNTELİK SEÇİM YÖNTEMİ	DEĞİŞKEN SAYISI	Zaman
IG+RFE	25	3,57 m	Kİ-KARE+RFE	25	4,30 m
IG+GA	25	11,5 m	Kİ-KARE +GA	36	10,9 m
IG+SA	13	1,93 m	Kİ-KARE +SA	18	1,85 m
IG+BORUTA	36	22,64 s	Kİ-KARE +BORUTA	39	21,80 s
GR+RFE	16	3,77 m	RELİEF+RFE	40	4,43 m
GR+GA	40	11,3 m	RELİEF +GA	33	14,7 m
GR+SA	11	1,93 m	RELİEF +SA	13	2,59 m
GR+BORUTA	25	21,99 s	RELİEF +BORUTA	25	36,5 s
CFS+RFE	12	43,8 s	SU+RFE	25	4,12 m
CFS +GA	5	9,32 m	SU+GA	27	17,6 m
CFS +SA	6	1,69 m	SU+SA	11	2,04 m
CFS+BORUTA	12	8,37 s	SU+BORUTA	33	27,54 s

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 15'te GlaucomaM veri seti için Temel algoritmalar, ardışık öznitelik seçimi sonrası seçilen nihai öznitelikler ile eğitilip tahmin yapılmıştır. Her bir algoritma için herhangi bir öznitelik seçiminin yapılmadığı durum (No Feature Selection) yani tüm öznitelikler kullanıldığında elde edilen sonuç ve ardışık öznitelik seçim yöntemleri sonucunda seçilen öznitelikler kullanılarak elde edilen sonuçlar verilmiştir. En iyi sonucu veren algoritma koyu olarak belirtilmiştir. Nerdeyse tüm algoritmalarda; öznitelik seçimi yapılmadığı duruma nazaran, ardışık öznitelik seçimi yapıldığında algoritmaların performansının daha iyi olduğu görülmektedir. Tüm sonuçlar incelendiğinde en iyi sonucu (Doğruluk= 95,24; Duyarlılık=92,5 ve Belirleyicilik=89,47) veren algoritmalar: RELİEF+RFE ardışık öznitelik seçimi sonrası MLP algoritması (5,4 m); SU+GA ardışık öznitelik seçimi sonrası C5.0 algoritması (17,51 m); IG+BO ardışık öznitelik seçimi sonrası Adaboost algoritması (9,28 m) ve

RELİEF+GA ardışık öznitelik seçimi sonrası Rotation Forest algoritması (17,2 m) dir. Aynı değerlere sahip olduğundan en iyi sonucu veren algoritma minimum hesaplama zamanına sahip olan RELİEF+RFE ardışık öznitelik seçimi sonrası MLP algoritmasıdır. Tablo 9 ile 15 kıyaslandığında; temel makine öğrenmesi algoritmalarında en yüksek doğruluk değeri %90 iken; ardışık öznitelik seçimi sonrası %95 değeri elde edilmiştir. Ancak herhangi bir öznitelik seçimi yapılmadığı durumdaki yığın algoritması ile kıyaslandığında yığın algoritması daha iyi bir performansa sahiptir.

**Tablo 15:** GlaucomaM Veri Seti için Ardışık Öznitelik Seçimi Sonrası Makine Öğrenmesi Algoritmalarının Sonuçları

ALGORİTMA	ÖZİNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
GLM	No Feature Selection	89,95	75	68,42
	IG+RFE	85,71	82,5	78,95
	IG+GA	76,19	82,5	89,47
	IG+SA	80,95	75	68,42
	IG+BORUTA	76,19	80	84,21
	GR+RFE	85,71	82,5	78,95
	GR+GA	76,19	77,5	78,95
	GR+SA	90,48	80	68,42
	GR+BORUTA	85,71	80	73,68
	CFS+RFE	90,48	80	68,42
	CFS +GA	90,48	85	78,95
	CFS +SA	80,95	75	68,42
	CFS+BORUTA	90,48	80	68,42
	Kİ-KARE+RFE	85,71	85	84,21
	Kİ-KARE +GA	85,71	80	73,68
	Kİ-KARE +SA	80,95	75	68,42
	Kİ-KARE +BORUTA	76,19	77,5	78,95
	RELİEF+RFE	80,95	85	89,47
	RELİEF +GA	76,19	77,5	78,95
	RELİEF +SA	85,71	80	73,68
	RELİEF +BORUTA	90,48	85	78,95
	SU+RFE	85,71	80	73,68
	SU+GA	66,67	77,5	89,47
	<b>SU+SA</b>	<b>90,48</b>	<b>87,5</b>	<b>84,21</b>
SU+BORUTA	76,19	80	84,21	
KNN	No Feature Selection	85,71	77,5	68,42
	IG+RFE	90,48	82,5	73,68
	IG+GA	80,95	72,5	63,16
	IG+SA	80,95	75	68,42
	IG+BORUTA	71,43	70	68,42
	GR+RFE	80,95	80	78,95
	GR+GA	80,95	72,5	63,16
	GR+SA	85,71	77,5	68,42
	GR+BORUTA	85,71	80	73,68
	CFS+RFE	85,71	80	73,68

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	CFS +GA	80,95	85	89,47
	CFS +SA	85,71	80	73,68
	CFS+BORUTA	85,71	80	73,68
	Kİ-KARE+RFE	85,71	80	73,68
	Kİ-KARE +GA	80,95	77,5	73,68
	Kİ-KARE +SA	85,71	77,5	68,42
	Kİ-KARE +BORUTA	76,19	72,5	68,42
	RELİEF+RFE	90,48	82,5	73,68
	<b>RELİEF +GA</b>	<b>90,48</b>	<b>85</b>	<b>78,95</b>
	RELİEF +SA	85,71	77,5	68,42
	RELİEF +BORUTA	85,71	82,5	78,95
	SU+RFE	90,48	82,5	73,68
	SU+GA	85,71	<b>87,5</b>	89,47
	SU+SA	80,95	80	78,95
	SU+BORUTA	76,19	75	73,68
NB	No Feature Selection	80,95	72,5	63,16
	IG+RFE	90,48	82,5	73,68
	IG+GA	85,71	72,5	57,89
	IG+SA	85,71	70	52,63
	IG+BORUTA	85,71	75	63,16
	GR+RFE	90,48	87,5	84,21
	GR+GA	85,71	72,5	57,89
	GR+SA	90,48	77,5	63,16
	GR+BORUTA	95,24	82,5	68,42
	<b>CFS+RFE</b>	<b>95,24</b>	<b>87,5</b>	<b>78,95</b>
	CFS +GA	90,48	<b>90</b>	89,47
	CFS +SA	85,71	80	73,68
	CFS+BORUTA	95,24	87,5	78,95
	Kİ-KARE+RFE	90,48	85	78,95
	Kİ-KARE +GA	85,71	75	63,16
	Kİ-KARE +SA	80,95	72,5	63,16
	Kİ-KARE +BORUTA	85,71	75	63,16
	RELİEF+RFE	80,95	70	57,89
	RELİEF +GA	80,95	70	57,89
	RELİEF +SA	80,95	72,5	63,16
	RELİEF +BORUTA	85,71	75	63,16
SU+RFE	90,48	80	68,42	
SU+GA	90,48	77,5	63,16	
SU+SA	90,48	87,5	84,21	
SU+BORUTA	85,71	75	63,16	
SVMLİNEAR	No Feature Selection	85,71	85	84,21
	IG+RFE	85,71	87,5	89,47
	IG+GA	85,71	87,5	89,47
	IG+SA	80,95	77,5	73,68
	IG+BORUTA	85,71	87,5	89,47
	GR+RFE	85,71	85	84,21
	GR+GA	85,71	82,5	78,95
	GR+SA	90,48	80	68,42
	GR+BORUTA	85,71	85	84,21
	CFS+RFE	85,71	82,5	78,95
	CFS +GA	90,48	85	78,95
	CFS +SA	80,95	77,5	73,68
	CFS+BORUTA	85,71	82,5	78,95

ALGORİTMA	ÖZNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Kİ-KARE+RFE	85,71	87,5	89,47
	Kİ-KARE +GA	85,71	80	73,68
	Kİ-KARE +SA	76,19	82,5	89,47
	Kİ-KARE +BORUTA	85,71	87,5	89,47
	RELİEF+RFE	80,95	82,5	84,21
	RELİEF +GA	80,95	80	78,95
	RELİEF +SA	76,19	77,5	78,95
	RELİEF +BORUTA	85,71	85	84,21
	SU+RFE	85,71	87,5	89,47
	SU+GA	80,95	85	89,47
	<b>SU+SA</b>	<b>90,48</b>	<b>87,5</b>	<b>84,21</b>
	SU+BORUTA	85,71	85	84,21
SVM POLY	No Feature Selection	90,48	90	89,47
	IG+RFE	80,95	82,5	84,21
	IG+GA	85,71	87,5	89,47
	IG+SA	85,71	80	73,68
	IG+BORUTA	85,71	85	84,21
	GR+RFE	76,19	80	84,21
	GR+GA	85,71	87,5	89,47
	GR+SA	95,24	82,5	68,42
	GR+BORUTA	80,95	82,5	84,21
	CFS+RFE	90,48	87,5	84,21
	CFS +GA	85,71	82,5	78,95
	CFS +SA	80,95	77,5	73,68
	CFS+BORUTA	90,48	87,5	84,21
	Kİ-KARE+RFE	80,95	85	89,47
	Kİ-KARE +GA	85,71	87,5	89,47
	Kİ-KARE +SA	85,71	80	73,68
	Kİ-KARE +BORUTA	85,71	87,5	89,47
	RELİEF+RFE	85,71	82,5	78,95
	RELİEF +GA	76,19	77,5	78,95
	RELİEF +SA	90,48	82,5	73,68
	RELİEF +BORUTA	85,71	85	84,21
	SU+RFE	80,95	82,5	84,21
	SU+GA	80,95	85	89,47
	<b>SU+SA</b>	<b>90,48</b>	<b>87,5</b>	<b>84,21</b>
SU+BORUTA	85,71	85	84,21	
SVM RADIAL	No Feature Selection	90,48	87,5	84,21
	IG+RFE	90,48	87,5	84,21
	IG+GA	90,48	85	78,95
	IG+SA	80,95	75	68,42
	IG+BORUTA	90,48	85	78,95
	GR+RFE	85,71	85	84,21
	GR+GA	85,71	82,5	78,95
	GR+SA	90,48	85	78,95
	GR+BORUTA	90,48	87,5	84,21
	CFS+RFE	80,95	82,5	84,21
	<b>CFS +GA</b>	<b>90,48</b>	<b>90</b>	<b>89,47</b>
	CFS +SA	80,95	80	78,95
	CFS+BORUTA	80,95	82,5	84,21
	Kİ-KARE+RFE	90,48	87,5	84,21
	Kİ-KARE +GA	90,48	87,5	84,21
	Kİ-KARE +SA	80,95	80	78,95

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Kİ-KARE +BORUTA	90,48	85	78,95
	RELİEF+RFE	90,48	87,5	84,21
	RELİEF +GA	90,48	87,5	84,21
	RELİEF +SA	85,71	80	73,68
	RELİEF +BORUTA	90,48	85	78,95
	SU+RFE	90,48	87,5	84,21
	SU+GA	85,71	85	84,21
	SU+SA	90,48	87,5	84,21
	SU+BORUTA	90,48	85	78,95
LDA	No Feature Selection	80,95	77,5	73,68
	IG+RFE	80,95	80	78,95
	IG+GA	76,19	77,5	78,95
	IG+SA	80,95	77,5	73,68
	IG+BORUTA	76,19	80	84,21
	GR+RFE	71,43	77,5	84,21
	GR+GA	80,95	82,5	84,21
	GR+SA	90,48	77,5	63,16
	GR+BORUTA	80,95	77,5	73,68
	CFS+RFE	80,95	77,5	73,68
	CFS +GA	90,48	80	68,42
	CFS +SA	80,95	75	68,42
	CFS+BORUTA	80,95	77,5	73,68
	Kİ-KARE+RFE	80,95	80	78,95
	Kİ-KARE +GA	80,95	80	78,95
	Kİ-KARE +SA	76,19	72,5	68,42
	Kİ-KARE +BORUTA	76,19	75	73,68
	RELİEF+RFE	80,95	85	89,47
	RELİEF +GA	66,67	72,5	78,95
	RELİEF +SA	76,19	75	73,68
	RELİEF +BORUTA	76,19	82,5	89,47
	SU+RFE	80,95	80	78,95
	SU+GA	76,19	80	84,21
<b>SU+SA</b>	<b>90,48</b>	<b>87,5</b>	<b>84,21</b>	
SU+BORUTA	71,43	77,5	84,21	
MLP	No Feature Selection	90,48	87,5	84,21
	IG+RFE	90,48	80	68,42
	IG+GA	80,95	87,5	94,74
	IG+SA	80,95	77,5	73,68
	IG+BORUTA	80,95	82,5	84,21
	GR+RFE	76,19	80	84,21
	GR+GA	71,43	80	89,47
	GR+SA	95,24	85	73,68
	GR+BORUTA	80,95	85	89,47
	CFS+RFE	85,71	80	73,68
	CFS +GA	85,71	72,5	57,89
	CFS +SA	85,71	80	73,68
	CFS+BORUTA	85,71	80	73,68
	Kİ-KARE+RFE	95,24	82,5	68,42
	Kİ-KARE +GA	95,24	75	52,63
	Kİ-KARE +SA	95,24	87,5	78,95
	Kİ-KARE +BORUTA	95,24	90	84,21
	<b>RELİEF+RFE</b>	<b>95,24</b>	<b>92,5</b>	<b>89,47</b>
	RELİEF +GA	90,48	87,5	84,21

ALGORİTMA	ÖZNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	RELİEF +SA	80,95	85	89,47
	RELİEF +BORUTA	95,24	87,5	78,95
	SU+RFE	90,48	80	68,42
	SU+GA	90,48	90	89,47
	SU+SA	85,71	87,5	89,47
	SU+BORUTA	76,19	82,5	89,47
EVTREE	No Feature Selection	66,67	80	94,74
	IG+RFE	76,19	77,5	78,95
	IG+GA	80,95	82,5	84,21
	IG+SA	90,48	82,5	73,68
	IG+BORUTA	76,19	77,5	78,95
	GR+RFE	76,19	77,5	78,95
	GR+GA	80,95	77,5	73,68
	GR+SA	80,95	85	89,47
	<b>GR+BORUTA</b>	<b>95,24</b>	<b>90</b>	84,21
	CFS+RFE	76,19	77,5	78,95
	CFS +GA	80,95	90	1
	CFS +SA	90,48	82,5	73,68
	CFS+BORUTA	76,19	82,5	89,47
	Kİ-KARE+RFE	76,19	82,5	89,47
	Kİ-KARE +GA	80,95	77,5	73,68
	Kİ-KARE +SA	95,24	85	73,68
	Kİ-KARE +BORUTA	76,19	77,5	78,95
	RELİEF+RFE	90,48	87,5	84,21
	RELİEF +GA	95,24	87,5	78,95
	RELİEF +SA	80,95	72,5	63,16
	RELİEF +BORUTA	95,24	82,5	68,42
	SU+RFE	76,19	77,5	78,95
	SU+GA	76,19	82,5	89,47
	SU+SA	71,43	77,5	84,21
SU+BORUTA	76,19	82,5	89,47	
RPART	No Feature Selection	90,48	80	68,42
	IG+RFE	90,48	80	68,42
	IG+GA	76,19	82,5	89,47
	IG+SA	90,48	80	68,42
	IG+BORUTA	90,48	80	68,42
	GR+RFE	90,48	80	68,42
	GR+GA	90,48	80	68,42
	GR+SA	85,71	77,5	68,42
	GR+BORUTA	90,48	80	68,42
	CFS+RFE	90,48	80	68,42
	CFS +GA	76,19	82,5	89,47
	CFS +SA	90,48	80	68,42
	CFS+BORUTA	90,48	80	68,42
	Kİ-KARE+RFE	90,48	80	68,42
	Kİ-KARE +GA	90,48	80	68,42
	Kİ-KARE +SA	90,48	80	68,42
	Kİ-KARE +BORUTA	90,48	80	68,42
	RELİEF+RFE	90,48	80	68,42
	RELİEF +GA	90,48	80	68,42
	RELİEF +SA	66,67	70	73,68
	RELİEF +BORUTA	90,48	80	68,42
	SU+RFE	90,48	80	68,42

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	SU+GA	90,48	80	68,42
	<b>SU+SA</b>	<b>90,48</b>	<b>87,5</b>	84,21
	SU+BORUTA	90,48	80	68,42
C5.0	No Feature Selection	80,95	80	78,95
	IG+RFE	95,24	90	84,21
	IG+GA	66,67	75	84,21
	IG+SA	76,19	80	84,21
	IG+BORUTA	85,71	87,5	89,47
	GR+RFE	85,71	80	73,68
	GR+GA	85,71	85	84,21
	GR+SA	95,24	80	63,16
	GR+BORUTA	90,48	85	78,95
	CFS+RFE	85,71	82,5	78,95
	CFS +GA	85,71	80	73,68
	CFS +SA	85,71	80	73,68
	CFS+BORUTA	85,71	82,5	78,95
	Kİ-KARE+RFE	95,24	85	73,68
	Kİ-KARE +GA	80,95	80	78,95
	Kİ-KARE +SA	85,71	80	73,68
	Kİ-KARE +BORUTA	85,71	85	84,21
	RELİEF+RFE	80,95	80	78,95
	RELİEF +GA	80,95	75	68,42
	RELİEF +SA	80,95	80	78,95
	RELİEF +BORUTA	90,48	82,5	73,68
	SU+RFE	85,71	85	84,21
	<b>SU+GA</b>	<b>95,24</b>	<b>92,5</b>	89,47
	SU+SA	80,95	85	89,47
SU+BORUTA	95,24	87,5	78,95	
J48	No Feature Selection	85,71	80	73,68
	IG+RFE	85,71	85	84,21
	IG+GA	76,19	77,5	78,95
	IG+SA	85,71	82,5	78,95
	IG+BORUTA	85,71	85	84,21
	GR+RFE	85,71	85	84,21
	GR+GA	80,95	85	89,47
	GR+SA	85,71	87,5	89,47
	GR+BORUTA	85,71	85	84,21
	CFS+RFE	85,71	75	63,16
	CFS +GA	80,95	77,5	73,68
	CFS +SA	85,71	75	63,16
	CFS+BORUTA	85,71	75	63,16
	Kİ-KARE+RFE	85,71	85	84,21
	Kİ-KARE +GA	85,71	85	84,21
	Kİ-KARE +SA	85,71	75	63,16
	Kİ-KARE +BORUTA	85,71	85	84,21
	RELİEF+RFE	80,95	85	89,47
	RELİEF +GA	66,67	77,5	89,47
	RELİEF +SA	76,19	75	73,68
	<b>RELİEF +BORUTA</b>	<b>95,24</b>	<b>85</b>	<b>73,68</b>
	SU+RFE	85,71	85	84,21
	SU+GA	85,71	85	84,21
	SU+SA	76,16	85	94,74
SU+BORUTA	85,71	85	84,21	

ALGORİTMA	ÖZNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
EARTH	No Feature Selection	80,95	82,5	84,21
	IG+RFE	80,95	82,5	84,21
	IG+GA	80,95	82,5	84,21
	IG+SA	85,71	80	73,68
	IG+BORUTA	80,95	82,5	84,21
	GR+RFE	76,19	77,5	78,95
	GR+GA	80,95	82,5	84,21
	GR+SA	80,95	80	78,95
	GR+BORUTA	80,95	82,5	84,21
	CFS+RFE	85,71	75	63,16
	CFS +GA	90,48	85	78,95
	CFS +SA	80,95	77,5	73,68
	CFS+BORUTA	85,71	75	63,16
	Kİ-KARE+RFE	80,95	82,5	84,21
	Kİ-KARE +GA	80,95	80	78,95
	Kİ-KARE +SA	85,71	85	84,21
	Kİ-KARE +BORUTA	80,95	82,5	84,21
	RELİEF+RFE	80,95	82,5	84,21
	RELİEF +GA	90,48	85	78,95
	<b>RELİEF +SA</b>	<b>95,24</b>	<b>85</b>	73,68
	RELİEF +BORUTA	80,95	82,5	84,21
	SU+RFE	80,95	82,5	84,21
	SU+GA	80,95	82,5	84,21
	SU+SA	80,95	82,5	84,21
SU+BORUTA	80,95	82,5	84,21	
RF	No Feature Selection	90,48	85	78,95
	IG+RFE	90,48	87,5	84,21
	IG+GA	85,71	90	94,74
	IG+SA	85,71	80	73,68
	IG+BORUTA	85,71	85	84,21
	GR+RFE	90,48	87,5	84,21
	GR+GA	90,48	85	78,95
	GR+SA	90,48	90	89,47
	GR+BORUTA	90,48	85	78,95
	CFS+RFE	85,71	82,5	78,95
	CFS +GA	90,48	87,5	84,21
	CFS +SA	90,48	77,5	63,16
	CFS+BORUTA	85,71	82,5	78,95
	Kİ-KARE+RFE	90,48	90	89,47
	<b>Kİ-KARE +GA</b>	<b>95,24</b>	<b>90</b>	84,21
	Kİ-KARE +SA	95,24	80	63,16
	Kİ-KARE +BORUTA	85,71	85	84,21
	RELİEF+RFE	90,48	82,5	73,68
	RELİEF +GA	95,24	85	73,68
	RELİEF +SA	90,48	<b>92,5</b>	94,74
	RELİEF +BORUTA	90,48	85	78,95
	SU+RFE	90,48	87,5	84,21
	SU+GA	90,48	87,5	84,21
	SU+SA	90,48	87,5	84,21
SU+BORUTA	85,71	85	84,21	
GBM	No Feature Selection	95,24	90	84,21
	IG+RFE	90,48	87,5	84,21
	IG+GA	80,95	82,5	84,21

ALGORİTMA	ÖZİNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	IG+SA	80,95	77,5	73,68
	IG+BORUTA	90,48	87,5	84,21
	GR+RFE	90,48	85	78,95
	GR+GA	90,48	87,5	84,21
	GR+SA	95,24	80	63,16
	GR+BORUTA	90,48	87,5	84,21
	CFS+RFE	85,71	82,5	78,95
	CFS +GA	90,48	85	78,95
	CFS +SA	76,19	77,5	78,95
	CFS+BORUTA	85,71	82,5	78,95
	Kİ-KARE+RFE	90,48	87,5	84,21
	Kİ-KARE +GA	90,48	87,5	84,21
	Kİ-KARE +SA	95,24	87,5	78,95
	<b>Kİ-KARE +BORUTA</b>	<b>95,24</b>	<b>90</b>	<b>84,21</b>
	RELİEF+RFE	90,48	90	89,47
	RELİEF +GA	95,24	85	73,68
	RELİEF +SA	85,71	87,5	89,47
	RELİEF +BORUTA	90,48	87,5	84,21
	SU+RFE	90,48	87,5	84,21
	SU+GA	90,48	90	89,47
SU+SA	85,71	85	84,21	
SU+BORUTA	90,48	87,5	84,21	
XGBTREE	No Feature Selection	85,71	85	84,21
	IG+RFE	90,48	90	89,47
	IG+GA	85,71	82,5	78,95
	IG+SA	90,48	80	68,42
	<b>IG+BORUTA</b>	<b>95,24</b>	<b>87,5</b>	<b>78,95</b>
	GR+RFE	90,48	<b>92,5</b>	94,74
	GR+GA	85,71	85	84,21
	GR+SA	90,48	87,5	84,21
	GR+BORUTA	85,71	85	84,21
	CFS+RFE	80,95	82,5	84,21
	CFS +GA	85,71	82,5	78,95
	CFS +SA	76,19	72,5	68,42
	CFS+BORUTA	85,71	85	84,21
	Kİ-KARE+RFE	90,48	87,5	84,21
	Kİ-KARE +GA	85,71	85	84,21
	Kİ-KARE +SA	95,24	85	73,68
	Kİ-KARE +BORUTA	90,48	85	78,95
	RELİEF+RFE	90,48	87,5	84,21
	RELİEF +GA	90,48	87,5	84,21
	RELİEF +SA	76,19	77,5	78,95
RELİEF +BORUTA	85,71	80	73,68	
SU+RFE	85,71	85	84,21	
SU+GA	85,71	85	84,21	
SU+SA	90,48	85	78,95	
SU+BORUTA	90,48	90	89,47	
TREEB AG	No Feature Selection	85,71	82,5	78,95
	IG+RFE	90,48	85	78,95
	IG+GA	85,71	87,5	89,47
	IG+SA	85,71	77,5	68,42
	IG+BORUTA	85,71	75	63,16
	GR+RFE	80,95	77,5	73,68

ALGORİTMA	ÖZİNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	GR+GA	85,71	82,5	78,95
	GR+SA	90,48	85	78,95
	GR+BORUTA	80,95	80	78,95
	CFS+RFE	80,95	80	78,95
	CFS +GA	85,71	87,5	89,47
	CFS +SA	95,24	80	63,16
	CFS+BORUTA	71,43	72,5	73,68
	Kİ-KARE+RFE	90,48	85	78,95
	Kİ-KARE +GA	90,48	87,5	84,21
	Kİ-KARE +SA	90,48	80	68,42
	Kİ-KARE +BORUTA	85,71	77,5	68,42
	RELİEF+RFE	85,71	75	63,16
	<b>RELİEF +GA</b>	<b>95,24</b>	<b>85</b>	<b>73,68</b>
	RELİEF +SA	90,48	87,5	84,21
	RELİEF +BORUTA	85,71	75	63,16
	SU+RFE	85,71	82,5	78,95
	SU+GA	90,48	87,5	84,21
	SU+SA	80,95	77,5	73,68
	SU+BORUTA	85,71	82,5	78,95
	ADABOOST	No Feature Selection	90,48	90
IG+RFE		90,48	87,5	84,21
IG+GA		85,71	87,5	89,47
IG+SA		85,71	77,5	68,42
<b>IG+BORUTA</b>		<b>95,24</b>	<b>92,5</b>	<b>89,47</b>
GR+RFE		85,71	85	84,21
GR+GA		90,48	87,5	84,21
GR+SA		90,48	80	68,42
GR+BORUTA		90,48	87,5	84,21
CFS+RFE		90,48	87,5	84,21
CFS +GA		90,48	87,5	84,21
CFS +SA		76,19	77,5	78,95
CFS+BORUTA		85,71	85	84,21
Kİ-KARE+RFE		85,71	85	84,21
Kİ-KARE +GA		90,48	87,5	84,21
Kİ-KARE +SA		95,24	85	73,68
Kİ-KARE +BORUTA		95,24	90	84,21
RELİEF+RFE		85,71	87,5	89,47
RELİEF +GA		95,24	90	84,21
RELİEF +SA		80,95	82,5	84,21
RELİEF +BORUTA		95,24	90	84,21
SU+RFE		85,71	85	84,21
SU+GA		85,71	87,5	89,47
SU+SA		80,95	82,5	84,21
SU+BORUTA	85,71	85	84,21	
GLMBOOST	No Feature Selection	80,95	77,5	73,68
	IG+RFE	80,95	77,5	73,68
	IG+GA	85,71	80	73,68
	IG+SA	85,71	77,5	68,42
	IG+BORUTA	80,95	77,5	73,68
	GR+RFE	80,95	75	68,42
	GR+GA	85,71	80	73,68
	<b>GR+SA</b>	<b>95,24</b>	<b>80</b>	<b>63,16</b>
	GR+BORUTA	80,95	80	78,95

ALGORİTMA	ÖZNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	CFS+RFE	85,71	80	73,68
	CFS +GA	90,48	82,5	73,68
	CFS +SA	80,95	75	68,42
	CFS+BORUTA	85,71	80	73,68
	Kİ-KARE+RFE	80,95	77,5	73,68
	Kİ-KARE +GA	85,71	80	73,68
	Kİ-KARE +SA	85,71	82,5	78,95
	Kİ-KARE +BORUTA	80,95	77,5	73,68
	RELİEF+RFE	80,95	75	68,42
	RELİEF +GA	80,95	75	68,42
	RELİEF +SA	76,19	75	73,68
	RELİEF +BORUTA	80,95	77,5	73,68
	SU+RFE	80,95	80	78,95
	SU+GA	85,71	77,5	68,42
	SU+SA	90,48	87,5	84,21
SU+BORUTA	80,95	77,5	73,68	
LOGİTBOOST	No Feature Selection	85,71	85	84,21
	IG+RFE	76,19	77,5	78,95
	IG+GA	71,43	80	89,47
	IG+SA	85,71	77,5	68,42
	IG+BORUTA	71,43	77,5	84,21
	GR+RFE	61,90	72,5	84,21
	GR+GA	80,95	82,5	84,21
	GR+SA	85,71	82,5	78,95
	GR+BORUTA	85,71	85	84,21
	CFS+RFE	76,19	80	84,21
	CFS +GA	85,71	80	73,68
	CFS +SA	66,67	77,5	89,47
	CFS+BORUTA	76,19	80	84,21
	Kİ-KARE+RFE	80,95	82,5	84,21
	Kİ-KARE +GA	76,19	82,5	89,47
	Kİ-KARE +SA	76,19	75	73,68
	Kİ-KARE +BORUTA	80,95	80	78,95
	RELİEF+RFE	80,95	80	78,95
	RELİEF +GA	85,71	82,5	78,95
	RELİEF +SA	85,71	80	73,68
	<b>RELİEF +BORUTA</b>	<b>85,71</b>	<b>87,5</b>	<b>89,47</b>
SU+RFE	76,19	77,5	78,95	
SU+GA	80,95	80	78,95	
SU+SA	80	95,24	63,16	
SU+BORUTA	80,95	82,5	84,21	
ROTATION FOREST	No Feature Selection	85,71	82,5	78,95
	IG+RFE	95,24	90	84,21
	IG+GA	85,71	82,5	78,95
	IG+SA	85,71	80	73,68
	IG+BORUTA	85,71	85	84,21
	GR+RFE	85,71	80	73,68
	GR+GA	90,48	87,5	84,21
	GR+SA	90,48	87,5	84,21
	GR+BORUTA	90,48	87,5	84,21
	CFS+RFE	85,71	80	73,68
	CFS +GA	85,71	87,5	89,47
	CFS +SA	85,71	77,5	68,42

ALGORİTMA	ÖZİNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	CFS+BORUTA	90,48	87,5	84,21
	Kİ-KARE+RFE	90,48	87,5	84,21
	Kİ-KARE +GA	90,48	87,5	84,21
	Kİ-KARE +SA	90,48	80	68,42
	Kİ-KARE +BORUTA	90,48	82,5	73,68
	RELİEF+RFE	90,48	85	78,95
	<b>RELİEF +GA</b>	<b>95,24</b>	<b>92,5</b>	<b>89,47</b>
	RELİEF +SA	71,43	75	78,95
	RELİEF +BORUTA	80,95	77,5	73,68
	SU+RFE	85,71	85	84,21
	SU+GA	90,48	85	78,95
	SU+SA	80,95	82,5	84,21
	SU+BORUTA	80,95	82,5	84,21

Kaynak: Yazar tarafından oluşturulmuştur.

Yapılan çalışmada, Bölüm 3.3.2 'de de değindiğimiz farklı temel model ve meta modeller kullanılarak 15504 model denenmiştir. Bu modellerin her birinin (15504), ardışık öznelik seçimi (24 seçim yöntemi) sonrası performansları incelenmiş. Yani toplamda 387.600 (15504\*24) ardışık öznelik seçim sonrası yığın modeli denenmiştir. Denenen modellere ilişkin sonuçların tamamına <https://tinyurl.com/5y793f62> linkinden erişilebilmektedir. Burada sadece belirli değerler üzerinde performans gösteren modeller verilmiştir. Tablo 16'da GlaucomaM veri seti için %95 ve üzeri doğruluk değerine sahip modeller verilmiştir.

Tablo 16'da kullanılan ardışık öznelik seçim yöntemi, kullanılan temel model, meta model ve performans ölçütlerinin değerleri (Doğruluk, Duyarlılık ve Belirleyicilik) verilmiştir. Doğruluk değeri dikkate alındığında; GR + SA ardışık öznelik seçimi sonrası yığın algoritması (Temel model: treebag, xgbTree, mlp; meta model: treebag; hesaplama zamanı: 9,3 m; değişken sayısı: 11) ve RELİEF + SA ardışık öznelik seçimi sonrası yığın algoritması (Temel model: knn, treebag, gbm; meta model: C5.0 hesaplama zamanı: 3,2 m; değişken sayısı: 13) en iyi sonucu vermiştir. Aynı Doğruluk değerlerine sahip olduklarından hesaplama zamanı dikkate alındığında RELİEF + SA ardışık öznelik seçimi sonrası yığın algoritması (Temel model: knn, treebag, gbm; meta model: C5.0) en iyi algoritmadır. Duyarlılık değeri dikkate alındığında; Duyarlılık değeri 100 olanların içerisinde minimum hesaplama zamanına sahip olan RELİEF + SA ardışık öznelik seçimi sonrası yığın algoritması (Temel model: C5.0, mlp, gbm; meta model: knn) en iyi sonucu vermiştir. Belirleyicilik değeri dikkate alındığında; Belirleyicilik değeri 100 olanların içerisinde minimum hesaplama zamanına sahip

olan IG+RFE ardışık öznelik seçimi sonrası yığın algoritması (Temel model: C5.0, lda, svmLinear; meta model: rotationForest) en iyi sonucu vermiştir.

**Tablo 16:** GlaucomaM Veri Seti için Ardışık Öznelik Seçimi Sonrası %95 Üzeri Doğruluk Veren Yığın Algoritması Sonuçları

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
IG+RFE	Model 542	knn	95	95,24	94,74
	Model 547	mlp	95	95,24	94,74
		rotationForest	95	90,48	100
		svmRadial	95	90,48	100
	Model 560	svmRadial	95	90,48	100
	Model 582	adaboost	95	95,24	94,74
		mlp	95	95,24	94,74
		rf	95	95,24	94,74
		svmRadial	95	95,24	94,74
	Model 632	adaboost	95	95,24	94,74
Model 643	earth	95	95,24	94,74	
IG+GA	Model 707	LogitBoost	95	95,24	94,74
		svmRadial	95	95,24	94,74
	Model 762	rotationForest	95	95,24	94,74
		svmRadial	95	95,24	94,74
IG+BORUTA	Model 167	svmLinear	95	95,24	94,74
	Model 258	LogitBoost	95	95,24	94,74
	Model 509	rotationForest	95	95,24	94,74
	Model 571	knn	95	95,24	94,74
	Model 599	glm	95	95,24	94,74
	Model 794	naive_bayes	95	95,24	94,74
GR+RFE	Model 585	C5.0	95	95,24	94,74
GR+GA	Model 161	glmboost	95	95,24	94,74
		svmLinear	95	95,24	94,74
		glm	95	95,24	94,74
	Model 428	xgbTree	95	100	89,47
	Model 511	svmPoly	95	95,24	94,74
	Model 735	xgbTree	95	95,24	94,74
	Model 767	adaboost	95	90,48	100
GR+SA	Model 52	svmPoly	95	95,24	94,74
	Model 388	treebag	<b>97,5</b>	<b>100</b>	<b>94,74</b>
		xgbTree	95	95,24	94,74
	Model 391	rf	95	100	89,47
	Model 396	rf	95	90,48	100
	Model 408	treebag	95	95,24	94,74
	Model 436	svmRadial	95	95,24	94,74
	Model 503	knn	95	95,24	94,74
	Model 721	svmRadial	95	95,24	94,74
CFS+GA	Model 343	rotationForest	95	90,48	100
	Model 678	adaboost	95	95,24	94,74
Kİ-KARE+RFE	Model 49	J48	95	95,24	94,74
	Model 54	svmPoly	95	95,24	94,74
	Model 144	J48	95	95,24	94,74
		rpart	95	95,24	94,74

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Model 174	glmboost	95	95,24	94,74
		glm	95	95,24	94,74
	Model 331	J48	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 466	glmboost	95	95,24	94,74
		rotationForest	95	95,24	94,74
		glm	95	95,24	94,74
	Model 475	naive_bayes	95	95,24	94,74
		glm	95	95,24	94,74
	Model 481	rotationForest	95	95,24	94,74
Model 527	rotationForest	95	95,24	94,74	
Kİ-KARE+GA	Model 115	C5.0	95	95,24	94,74
	Model 167	earth	95	95,24	94,74
	Model 486	adaboost	95	95,24	94,74
		rf	95	95,24	94,74
	Model 577	rotationForest	95	95,24	94,74
Kİ-KARE+SA	Model 228	LogitBoost	95	95,24	94,74
	Model 617	C5.0	95	95,24	94,74
		J48	95	95,24	94,74
		earth	95	95,24	94,74
	Model 698	adaboost	95	95,24	94,74
Kİ-KARE+BORUTA	Model 231	adaboost	95	95,24	94,74
	Model 246	rotationForest	95	95,24	94,74
	Model 382	knn	95	95,24	94,74
	Model 461	knn	95	95,24	94,74
	Model 633	LogitBoost	95	95,24	94,74
		xgbTree	95	95,24	94,74
	Model 707	earth	95	95,24	94,74
RELİEF+RFE	Model 18	naive_bayes	95	95,24	94,74
	Model 95	adaboost	95	95,24	94,74
	Model 121	glmboost	95	95,24	94,74
		lda	95	95,24	94,74
		naive_bayes	95	95,24	94,74
		svmLinear	95	95,24	94,74
		svmPoly	95	95,24	94,74
		glm	95	95,24	94,74
	Model 214	rpart	95	95,24	94,74
	Model 219	glmboost	95	95,24	94,74
		lda	95	95,24	94,74
		earth	95	95,24	94,74
		svmLinear	95	95,24	94,74
		svmPoly	95	95,24	94,74
	Model 220	rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 243	earth	95	95,24	94,74
	Model 245	earth	95	95,24	94,74
	Model 470	knn	95	95,24	94,74
	Model 519	adaboost	95	95,24	94,74
	Model 520	rf	95	95,24	94,74
		rotationForest	95	95,24	94,74
	Model 631	svmRadial	95	95,24	94,74
Model 657	knn	95	95,24	94,74	

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Model 681	rotationForest	95	90,48	100
	Model 685	adaboost	95	95,24	94,74
	Model 698	rpart	95	95,24	94,74
	Model 701	rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 702	rpart	95	95,24	94,74
	Model 703	rpart	95	95,24	94,74
	Model 703	J48	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 705	C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 706	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 707	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 708	J48	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 709	rpart	95	95,24	94,74
	Model 710	rpart	95	95,24	94,74
		xgbTree	95	95,24	94,74
	Model 711	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		xgbTree	95	95,24	94,74
	Model 712	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 713	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 714	J48	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 715	J48	95	95,24	94,74
		earth	95	95,24	94,74
	Model 716	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 717	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 718	J48	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 719	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 720	C5.0	95	95,24	94,74
		rpart	95	95,24	94,74

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Model 721	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 722	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 723	earth	95	95,24	94,74
		rf	95	95,24	94,74
	Model 724	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
	Model 725	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		knn	95	95,24	94,74
		earth	95	95,24	94,74
		rf	95	95,24	94,74
	Model 726	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
		rf	95	95,24	94,74
	Model 727	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		rpart	95	95,24	94,74
	Model 728	J48	95	95,24	94,74
		rpart	95	95,24	94,74
		earth	95	95,24	94,74
Model 729	J48	95	95,24	94,74	
	C5.0	95	95,24	94,74	
	rpart	95	95,24	94,74	
	earth	95	95,24	94,74	
Model 730	adaboost	95	95,24	94,74	
	rpart	95	95,24	94,74	
	xgbTree	95	95,24	94,74	
	rf	95	95,24	94,74	
	svmRadial	95	95,24	94,74	
Model 731	adaboost	95	95,24	94,74	
	J48	95	95,24	94,74	
	C5.0	95	95,24	94,74	
	xgbTree	95	95,24	94,74	
	knn	95	95,24	94,74	
	earth	95	95,24	94,74	
Model 732	J48	95	95,24	94,74	
	C5.0	95	95,24	94,74	
	rpart	95	95,24	94,74	
	earth	95	95,24	94,74	
RELİEF+GA	Model 11	rpart	95	95,24	94,74
	Model 49	svmPoly	95	95,24	94,74
	Model 319	rotationForest	95	95,24	94,74
	Model 585	C5.0	95	95,24	94,74

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Model 699	rf	95	95,24	94,74
	Model 806	LogitBoost	95	95,24	94,74
RELİEF+SA	Model 36	earth	95	95,24	94,74
	Model 38	earth	95	95,24	94,74
	Model 40	C5.0	<b>97,5</b>	<b>95,24</b>	<b>100</b>
	Model 47	xgbTree	95	95,24	94,74
		knn	95	95,24	94,74
	Model 49	rf	95	95,24	94,74
		rotationForest	95	95,24	94,74
		svmRadial	95	95,24	94,74
	Model 109	glmboost	95	95,24	94,74
		xgbTree	95	95,24	94,74
		lda	95	95,24	94,74
		earth	95	95,24	94,74
		svmLinear	95	95,24	94,74
		glm	95	95,24	94,74
	Model 111	knn	95	95,24	94,74
		rotationForest	95	95,24	94,74
	Model 114	glmboost	95	95,24	94,74
	Model 144	treebag	95	95,24	94,74
		rf	95	95,24	94,74
	Model 171	mlp	95	95,24	94,74
	Model 194	mlp	95	95,24	94,74
		glm	95	95,24	94,74
	Model 218	C5.0	95	95,24	94,74
		knn	95	95,24	94,74
	Model 219	glmboost	95	95,24	94,74
		lda	95	95,24	94,74
	Model 229	glmboost	95	95,24	94,74
		rpart	95	95,24	94,74
		lda	95	95,24	94,74
	Model 234	treebag	95	90,48	100
	Model 410	naive_bayes	95	95,24	94,74
	Model 426	lda	95	95,24	94,74
	Model 487	earth	95	95,24	94,74
	Model 504	svmLinear	95	95,24	94,74
	Model 616	svmPoly	95	90,48	100
	Model 617	svmPoly	95	90,48	100
		svmRadial	95	90,48	100
	Model 618	mlp	95	90,48	100
		svmLinear	95	90,48	100
		svmRadial	95	90,48	100
	Model 619	rpart	95	90,48	100
	Model 622	rotationForest	95	90,48	100
svmPoly		95	90,48	100	
Model 624	glmboost	95	95,24	94,74	
	lda	95	95,24	94,74	
	mlp	95	95,24	94,74	
	svmPoly	95	90,48	100	
	glm	95	95,24	94,74	
Model 625	glmboost	95	95,24	94,74	
	xgbtree	95	90,48	100	
	mlp	95	95,24	94,74	

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Model 629	naive_bayes	95	90,48	100
	Model 631	rpart	95	90,48	100
	Model 635	glm	95	90,48	100
	Model 701	rotationForest	95	95,24	94,74
	Model 705	earth	95	95,24	94,74
	Model 706	glmboost	95	95,24	94,74
		lda	95	95,24	94,74
	Model 709	svmPoly	95	95,24	94,74
	Model 714	svmPoly	95	95,24	94,74
	Model 727	rotationForest	95	95,24	94,74
svmPoly		95	95,24	94,74	
RELİEF+BORUTA	Model 72	adaboost	95	95,24	94,74
	Model 115	treebag	95	95,24	94,74
	Model 470	knn	95	95,24	94,74
		mlp	95	95,24	94,74
	Model 480	earth	95	95,24	94,74
	Model 489	rotationForest	95	95,24	94,74
		svmRadial	95	95,24	94,74
	Model 497	earth	95	95,24	94,74
	Model 555	knn	95	<b>100</b>	89,47
	Model 585	LogitBoost	95	95,24	94,74
Model 699	svmPoly	95	95,24	94,74	
SU+RFE	Model 54	rf	95	<b>100</b>	89,47
	Model 802	knn	95	95,24	94,74
SU+GA	Model 48	adaboost	95	95,24	94,74
	Model 51	svmRadial	95	95,24	94,74
	Model 66	rotationForest	95	95,24	94,74
		svmPoly	95	95,24	94,74
	Model 94	J48	95	95,24	94,74
	Model 111	glmboost	95	95,24	94,74
		svmLinear	95	95,24	94,74
		glm	95	95,24	94,74
	Model 181	glmboost	95	95,24	94,74
	Model 231	naive_bayes	95	95,24	94,74
	Model 235	naive_bayes	95	95,24	94,74
	Model 398	mlp	95	95,24	94,74
		svmLinear	95	95,24	94,74
	Model 403	C5.0	95	90,48	100
	Model 409	glmboost	95	95,24	94,74
		C5.0	95	95,24	94,74
		lda	95	95,24	94,74
		svmLinear	95	95,24	94,74
	Model 412	lda	95	95,24	94,74
		naive_bayes	95	95,24	94,74
	Model 414	svmRadial	95	95,24	94,74
		xgbtree	95	95,24	94,74
	Model 415	earth	95	95,24	94,74
		lda	95	95,24	94,74
	Model 416	svmLinear	95	95,24	94,74
		glmboost	95	95,24	94,74
		naive_bayes	95	95,24	94,74
	Model 427	glm	95	95,24	94,74
svmLinear		95	95,24	94,74	

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Model 433	lda	95	95,24	94,74
		mlp	95	95,24	94,74
		svmLinear	95	95,24	94,74
		svmPoly	95	95,24	94,74
		svmRadial	95	95,24	94,74
	Model 443	glmboost	95	95,24	94,74
		lda	95	95,24	94,74
		mlp	95	95,24	94,74
		svmLinear	95	95,24	94,74
		glm	95	95,24	94,74
	Model 444	mlp	95	95,24	94,74
		svmPoly	95	95,24	94,74
	Model 446	svmRadial	95	95,24	94,74
	Model 455	rotationForest	95	95,24	94,74
	Model 490	naive_bayes	95	95,24	94,74
		rotationForest	95	95,24	94,74
		svmRadial	95	95,24	94,74
	Model 762	glm	95	95,24	94,74
	Model 767	lda	95	95,24	94,74
		svmLinear	95	95,24	94,74
Model 771	glm	95	95,24	94,74	
Model 775	glmboost	95	95,24	94,74	
	lda	95	95,24	94,74	
SU+SA	Model 162	adaboost	95	100	89,47
SU+BORUTA	Model 175	naive_bayes	95	95,24	94,74
	Model 319	adaboost	95	95,24	94,74
	Model 360	J48	95	95,24	94,74
		C5.0	95	95,24	94,74
		xgbtree	95	95,24	94,74
	Model 604	xgbtree	95	95,24	94,74
	Model 643	xgbtree	95	95,24	94,74
	Model 645	adaboost	95	95,24	94,74

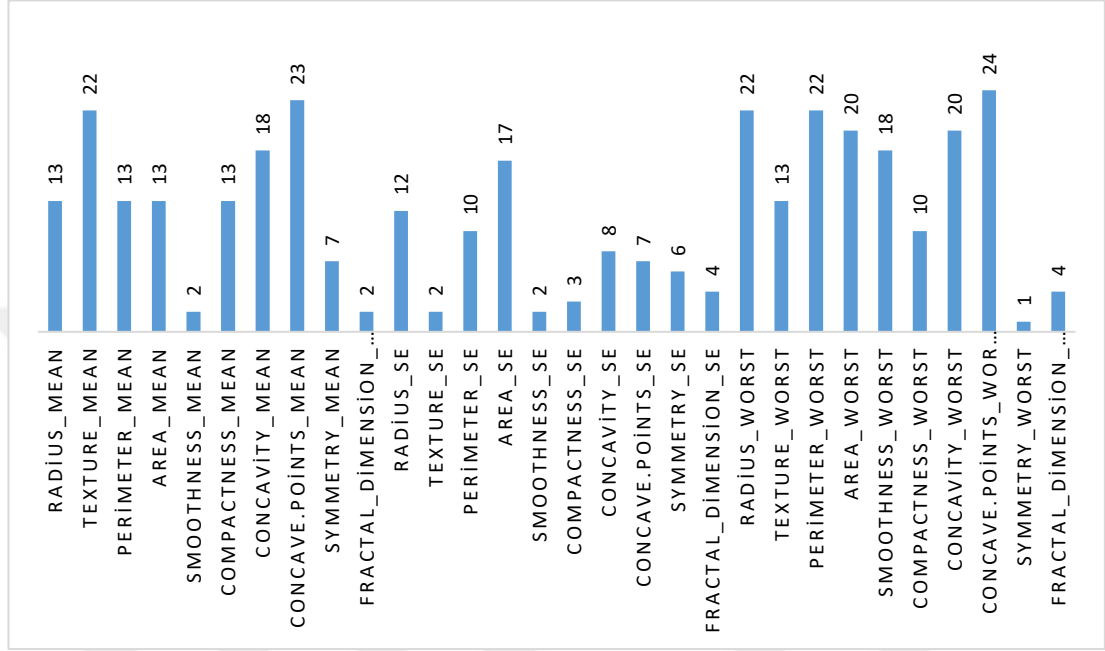
Kaynak: Yazar tarafından oluşturulmuştur.

### 3.6.1. WDBC Veri Setinin Ardışık Öznitelik Seçimi Sonuçları

WDBC veri setinde 30 tanımlayıcı özellik ve bir adet teşhis sınıfı (iyi huylu veya kötü huylu) olmak üzere toplam 31 öznitelik bulunmaktadır. Her bir öznitelik seçim yöntemi tarafından seçilen değişkenler dikkate alınarak seçilme sıklıkları elde edilmiştir. Öznitelik seçimi sonrası değişkenlerin seçilme sıklıkları Şekil 32' de verilmiştir. Şekil incelendiğinde; concave.points\_worst, concave.points\_mean, texture\_mean, radius\_worst, perimeter\_worst, area\_worst ve concavity\_worst öznitelikleri yirmi ve üzeri defa seçilmiştir yani en önemli değişkenlerdir. symmetry\_worst, smoothness\_mean, fractal\_dimension\_mean, texture\_se ve

smoothness\_se öznitelikleri ise iki ve daha az defa seçildiğinden en önemsiz değişkenlerdir.

**Şekil 32:** WDBC Veri Seti için Değişkenlerin Seçilme Sayısı



Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 17’ de ardışık öznitelik seçimi sonrası seçilen değişkenleri sayıları ve hesaplama zamanları verilmiştir. Değişken sayımız ne kadar azsa, algoritmaların karmaşıklığı azalarak anlaşılabilirliği artmaktadır aynı zamanda öğrenme süresi de kısaltmakta algoritmaların hızı artmaktadır. Ayrıca önemsiz, ilgisiz, gürültülü ve yanıltıcı değişkenlerin çıkarılması algoritmaların performansını önemli ölçüde artırmaktadır. Tablo 17 incelendiğinde, en az değişkene IG ardından SA uygulanmış ve CFS ardından SA uygulanmış hibrit öznitelik yaklaşımları; en fazla değişkene ise Kİ-KARE ardından BORUTA uygulanmış hibrit öznitelik yaklaşımının sahip olduğu görülmektedir. Ayrıca, hesaplama zamanı açısından incelediğimizde en az süreye IG ardından BORUTA uygulanmış hibrit öznitelik yaklaşımı; en fazla süreye ise Kİ-KARE ardından GA uygulanmış hibrit öznitelik yaklaşımının sahip olduğu görülmektedir

**Tablo 17:** WDBC Veri Seti için Ardışık Öznitelik Seçimi ile Seçilen Değişken Sayıları ve Hesaplama Zamanı

ÖZİNİTELİK SEÇİM YÖNTEMİ	DEĞİŞKEN SAYISI	Zaman	ÖZİNİTELİK SEÇİM YÖNTEMİ	DEĞİŞKEN SAYISI	Zaman
IG+RFE	7	2,7 m	Kİ-KARE+RFE	15	4,5 m
IG+GA	6	16,5 m	Kİ-KARE +GA	15	25,5 m
IG+SA	4	3,9 m	Kİ-KARE +SA	11	4,2 m
IG+BORUTA	10	2,2 s	Kİ-KARE +BORUTA	26	42,7 s
GR+RFE	10	2,9 m	RELİEF+RFE	15	2,5 m
GR+GA	9	18,8 m	RELİEF +GA	12	11,9 m
GR+SA	8	3,7 m	RELİEF +SA	12	2,4 m
GR+BORUTA	20	34,2 s	RELİEF +BORUTA	20	46,7 s
CFS+RFE	9	1,8 m	SU+RFE	20	2,3 m
CFS +GA	7	17,8 m	SU+GA	10	11,7 m
CFS +SA	4	3,3 m	SU+SA	9	2,1 m
CFS+BORUTA	9	3,4 s	SU+BORUTA	20	4,01 s

Kaynak: Yazar tarafından oluşturulmuştur.

Tablo 18’de WDBC veri seti için temel algoritmalar, ardışık öznitelik seçimi sonrası seçilen nihai öznitelikler ile eğitilip tahmin yapılmıştır. Her bir algoritma için herhangi bir öznitelik seçiminin yapılmadığı durum (No Feature Selection) yani tüm öznitelikler kullanıldığında elde edilen sonuç ve ardışık öznitelik seçim yöntemleri sonucunda seçilen öznitelikler kullanılarak elde edilen sonuçlar verilmiştir. En iyi sonucu veren algoritma koyu olarak belirtilmiştir. Nerdeyse tüm algoritmalarda; öznitelik seçimi yapılmadığı duruma nazaran, ardışık öznitelik seçimi yapıldığında algoritmaların performansının daha iyi olduğu görülmektedir. Tüm sonuçlar incelendiğinde en iyi sonucu Kİ-KARE+BORUTA ardışık öznitelik seçimi sonrası SvmRadial algoritması en iyi sonucu vermiştir.

**Tablo 18:** WDBC Veri Seti için Ardışık Öznitelik Seçimi Sonrası Makine Öğrenmesi Algoritmalarının Sonuçları

ALGORİTMA	ÖZİNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
GLM	No Feature Selection	93,86	97,18	88,37
	IG+RFE	92,98	95,77	88,37
	IG+GA	93,86	97,18	88,37
	IG+SA	92,11	95,77	86,05
	IG+BORUTA	95,61	98,59	90,70
	GR+RFE	93,86	97,18	88,37
	GR+GA	92,98	95,77	88,37
	<b>GR+SA</b>	<b>95,61</b>	<b>100</b>	<b>88,37</b>

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	GR+BORUTA	94,74	97,18	90,70
	CFS+RFE	93,86	97,18	88,37
	CFS +GA	94,74	98,59	88,37
	CFS +SA	92,11	97,18	83,72
	CFS+BORUTA	93,86	97,18	88,37
	Kİ-KARE+RFE	95,61	97,18	93,02
	Kİ-KARE +GA	93,86	98,59	86,05
	Kİ-KARE +SA	93,86	97,18	88,37
	Kİ-KARE +BORUTA	95,61	98,59	90,70
	RELİEF+RFE	94,74	98,59	88,37
	RELİEF +GA	92,98	97,18	86,05
	RELİEF +SA	93,86	97,18	88,37
	RELİEF +BORUTA	95,61	98,59	90,70
	SU+RFE	94,74	97,18	90,70
	SU+GA	94,74	98,59	88,37
	SU+SA	93,86	98,59	86,05
	SU+BORUTA	94,74	97,18	90,70
KNN	No Feature Selection	92,98	100	81,39
	IG+RFE	92,11	98,59	81,40
	IG+GA	90,35	98,59	76,74
	IG+SA	92,11	97,18	83,72
	IG+BORUTA	92,98	100	81,39
	GR+RFE	92,11	98,59	81,40
	GR+GA	91,23	97,18	81,40
	GR+SA	92,11	98,59	81,40
	GR+BORUTA	92,98	100	81,39
	CFS+RFE	92,11	98,59	81,40
	CFS +GA	91,23	97,18	81,40
	CFS +SA	92,11	95,77	86,05
	CFS+BORUTA	92,11	98,59	81,40
	Kİ-KARE+RFE	92,98	100	81,39
	Kİ-KARE +GA	92,98	98,59	83,72
	Kİ-KARE +SA	92,11	92,96	90,70
	Kİ-KARE +BORUTA	92,98	100	81,39
	RELİEF+RFE	92,98	100	81,39
	RELİEF +GA	92,11	97,18	83,72
	RELİEF +SA	92,98	100	81,39
	RELİEF +BORUTA	92,98	100	81,39
SU+RFE	92,98	100	81,39	
SU+GA	92,98	100	81,39	
<b>SU+SA</b>	<b>93,86</b>	<b>100</b>	<b>83,72</b>	
SU+BORUTA	92,98	100	81,39	
NB	No Feature Selection	92,98	95,77	88,37
	IG+RFE	94,74	97,18	90,70
	IG+GA	93,86	98,59	86,05
	IG+SA	93,86	97,18	88,37
	IG+BORUTA	92,98	98,59	83,72
	GR+RFE	94,74	97,18	90,70
	GR+GA	92,98	97,18	86,05
	GR+SA	92,98	98,59	83,72
	GR+BORUTA	92,98	95,77	88,37
	CFS+RFE	94,74	97,18	90,70
	CFS +GA	92,11	94,37	88,37

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	CFS +SA	92,98	95,77	88,37
	CFS+BORUTA	94,74	97,18	90,70
	Kİ-KARE+RFE	94,74	100	86,05
	<b>Kİ-KARE +GA</b>	<b>95,61</b>	<b>100</b>	<b>88,37</b>
	Kİ-KARE +SA	92,11	94,37	88,37
	Kİ-KARE +BORUTA	92,98	95,77	88,37
	RELİEF+RFE	94,74	98,59	88,37
	RELİEF +GA	93,86	98,59	86,05
	RELİEF +SA	90,35	92,96	86,05
	RELİEF +BORUTA	93,86	97,18	88,37
	SU+RFE	92,98	95,77	88,37
	SU+GA	94,74	98,59	88,37
	SU+SA	94,74	100	86,05
	SU+BORUTA	92,98	95,77	88,37
SVMLİNEAR	No Feature Selection	97,37	100	93,02
	IG+RFE	92,98	97,18	86,05
	IG+GA	94,74	98,59	88,37
	IG+SA	90,35	95,77	81,40
	IG+BORUTA	92,98	98,59	83,72
	GR+RFE	94,74	100	86,05
	GR+GA	95,61	100	88,37
	<b>GR+SA</b>	<b>95,61</b>	<b>100</b>	<b>88,37</b>
	GR+BORUTA	95,61	100	88,37
	CFS+RFE	93,86	98,59	86,05
	CFS +GA	92,98	97,18	86,05
	CFS +SA	91,23	98,59	79,07
	CFS+BORUTA	93,86	98,59	86,05
	Kİ-KARE+RFE	94,74	98,59	88,37
	Kİ-KARE +GA	94,74	98,59	88,37
	Kİ-KARE +SA	93,86	97,18	88,37
	Kİ-KARE +BORUTA	95,61	98,59	90,70
	RELİEF+RFE	93,86	98,59	86,05
	RELİEF +GA	94,74	98,59	88,37
	RELİEF +SA	93,86	98,59	86,05
	RELİEF +BORUTA	95,61	100	88,37
	SU+RFE	93,86	98,59	86,05
	SU+GA	94,74	98,59	88,37
	SU+SA	94,74	98,59	88,37
SU+BORUTA	93,86	98,59	86,05	
SVMPOLY	No Feature Selection	96,49	100	90,70
	IG+RFE	92,98	98,59	83,72
	IG+GA	92,98	98,59	83,72
	IG+SA	91,23	98,59	79,07
	IG+BORUTA	91,23	98,59	79,07
	GR+RFE	94,74	100	86,05
	GR+GA	93,86	100	83,72
	GR+SA	94,74	100	86,05
	GR+BORUTA	95,61	100	88,37
	CFS+RFE	93,86	98,59	86,05
	CFS +GA	93,86	98,59	86,05
	CFS +SA	90,35	98,59	76,74
	CFS+BORUTA	93,86	98,59	86,05
	Kİ-KARE+RFE	95,61	98,59	90,70

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	Kİ-KARE +GA	95,61	100	88,37
	Kİ-KARE +SA	92,98	95,77	88,37
	<b>Kİ-KARE +BORUTA</b>	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	RELİEF+RFE	95,61	98,59	90,70
	RELİEF +GA	94,74	98,59	88,37
	RELİEF +SA	92,98	98,59	83,72
	RELİEF +BORUTA	95,61	100	88,37
	SU+RFE	94,74	98,59	88,37
	SU+GA	95,61	98,59	90,70
	SU+SA	94,74	98,59	88,37
	SU+BORUTA	94,74	98,59	88,37
	SVMRADIAL	No Feature Selection	97,37	100
IG+RFE		94,74	98,59	88,37
IG+GA		94,74	98,59	88,37
IG+SA		93,86	98,59	86,05
IG+BORUTA		95,61	98,59	90,70
GR+RFE		94,74	100	86,05
GR+GA		95,61	100	88,37
GR+SA		95,61	100	88,37
GR+BORUTA		96,49	100	90,70
CFS+RFE		94,74	100	86,05
CFS +GA		96,49	100	90,70
CFS +SA		92,11	94,37	88,37
CFS+BORUTA		94,74	100	86,05
Kİ-KARE+RFE		95,61	98,59	90,70
Kİ-KARE +GA		95,61	100	88,37
Kİ-KARE +SA		93,86	95,77	90,70
<b>Kİ-KARE +BORUTA</b>		<b>98,25</b>	<b>100</b>	<b>95,35</b>
RELİEF+RFE		95,61	100	88,37
RELİEF +GA		97,37	100	93,02
RELİEF +SA		93,86	98,59	86,05
RELİEF +BORUTA		97,37	100	93,02
SU+RFE		96,49	100	90,70
SU+GA		95,61	97,18	93,02
SU+SA		95,61	100	88,37
SU+BORUTA	96,49	100	90,70	
LDA	No Feature Selection	<b>97,74</b>	<b>100</b>	<b>86,05</b>
	IG+RFE	92,98	98,59	83,72
	IG+GA	92,98	100	81,40
	IG+SA	92,98	100	81,40
	IG+BORUTA	92,98	100	81,40
	<b>GR+RFE</b>	<b>94,74</b>	<b>100</b>	<b>86,05</b>
	GR+GA	93,86	100	83,72
	GR+SA	93,86	100	83,72
	GR+BORUTA	92,98	100	81,40
	CFS+RFE	92,98	100	81,40
	CFS +GA	92,11	100	79,07
	CFS +SA	90,35	98,59	76,74
	CFS+BORUTA	92,98	100	81,40
	Kİ-KARE+RFE	93,86	100	83,72
	Kİ-KARE +GA	94,74	100	86,05
	Kİ-KARE +SA	91,23	97,18	81,40
	Kİ-KARE +BORUTA	94,74	100	86,05

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	RELİEF+RFE	94,74	100	86,05
	RELİEF +GA	93,86	100	83,72
	RELİEF +SA	92,98	98,59	83,72
	RELİEF +BORUTA	93,86	100	83,72
	SU+RFE	92,98	100	81,40
	SU+GA	91,23	98,59	79,07
	SU+SA	92,11	100	79,07
	SU+BORUTA	92,98	100	81,40
MLP	No Feature Selection	62,28	100	0
	IG+RFE	62,28	100	0
	IG+GA	62,28	100	0
	IG+SA	62,28	100	0
	IG+BORUTA	62,28	100	0
	GR+RFE	62,28	100	0
	GR+GA	62,28	100	0
	GR+SA	62,28	100	0
	GR+BORUTA	62,28	100	0
	CFS+RFE	62,28	100	0
	CFS +GA	62,28	100	0
	CFS +SA	62,28	100	0
	CFS+BORUTA	62,28	100	0
	Kİ-KARE+RFE	62,28	100	0
	Kİ-KARE +GA	62,28	100	0
	<b>Kİ-KARE +SA</b>	<b>92,11</b>	<b>95,77</b>	<b>86,05</b>
	Kİ-KARE +BORUTA	62,28	100	0
	RELİEF+RFE	62,28	100	0
	RELİEF +GA	62,28	100	0
	RELİEF +SA	62,28	100	0
RELİEF +BORUTA	62,28	100	0	
SU+RFE	62,28	100	0	
SU+GA	62,28	100	0	
SU+SA	62,28	100	0	
SU+BORUTA	62,28	100	0	
EVTREE	No Feature Selection	94,74	98,59	88,37
	IG+RFE	92,98	97,18	86,05
	IG+GA	92,98	97,18	86,05
	IG+SA	91,23	97,18	81,40
	IG+BORUTA	92,98	97,18	86,05
	GR+RFE	94,74	98,59	88,37
	GR+GA	94,74	98,59	88,37
	GR+SA	94,74	98,59	88,37
	GR+BORUTA	92,98	97,18	86,05
	CFS+RFE	92,98	97,18	86,05
	CFS +GA	94,74	98,59	88,37
	CFS +SA	91,23	95,77	83,72
	CFS+BORUTA	92,98	97,18	86,05
	Kİ-KARE+RFE	93,86	97,18	88,37
	Kİ-KARE +GA	93,86	97,18	88,37
	Kİ-KARE +SA	89,47	91,55	86,05
	Kİ-KARE +BORUTA	92,98	97,18	86,05
	RELİEF+RFE	92,98	97,18	86,05
	RELİEF +GA	92,11	95,77	86,05
	RELİEF +SA	92,98	97,18	86,05

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	RELİEF +BORUTA	92,98	97,18	86,05
	SU+RFE	92,98	97,18	86,05
	<b>SU+GA</b>	<b>96,49</b>	<b>98,59</b>	<b>93,02</b>
	SU+SA	93,86	97,18	88,37
	SU+BORUTA	94,74	98,59	88,37
RPART	No Feature Selection	91,22	95,77	83,72
	IG+RFE	91,22	95,77	83,72
	<b>IG+GA</b>	<b>93,86</b>	<b>97,18</b>	<b>88,37</b>
	IG+SA	88,60	95,77	76,74
	IG+BORUTA	91,22	95,77	83,72
	GR+RFE	91,22	95,77	83,72
	GR+GA	93,86	97,18	88,37
	GR+SA	92,98	98,59	83,72
	GR+BORUTA	91,22	95,77	83,72
	CFS+RFE	91,22	95,77	83,72
	CFS +GA	93,86	97,18	88,37
	CFS +SA	92,98	98,59	83,72
	CFS+BORUTA	91,22	95,77	83,72
	Kİ-KARE+RFE	91,22	95,77	83,72
	Kİ-KARE +GA	92,98	98,59	83,72
	Kİ-KARE +SA	89,47	91,55	86,05
	Kİ-KARE +BORUTA	91,22	95,77	83,72
	RELİEF+RFE	91,22	95,77	83,72
	RELİEF +GA	92,11	97,18	83,72
	RELİEF +SA	92,11	97,18	83,72
	RELİEF +BORUTA	91,22	95,77	83,72
	SU+RFE	91,22	95,77	83,72
	SU+GA	92,98	97,18	86,05
	SU+SA	<b>93,86</b>	<b>97,18</b>	<b>88,37</b>
SU+BORUTA	91,22	95,77	83,72	
C5.0	No Feature Selection	95,61	100	88,37
	IG+RFE	92,98	95,77	88,37
	IG+GA	94,74	97,18	90,70
	IG+SA	93,86	95,77	90,70
	IG+BORUTA	92,98	95,77	88,37
	GR+RFE	95,61	98,59	90,70
	GR+GA	94,74	97,18	90,70
	GR+SA	95,61	100	88,37
	<b>GR+BORUTA</b>	<b>97,37</b>	<b>100</b>	<b>93,02</b>
	CFS+RFE	96,49	100	90,70
	CFS +GA	96,49	100	90,70
	CFS +SA	89,47	92,96	83,72
	CFS+BORUTA	96,49	100	90,70
	Kİ-KARE+RFE	95,61	100	88,37
	Kİ-KARE +GA	95,61	100	88,37
	Kİ-KARE +SA	92,11	94,37	88,37
	Kİ-KARE +BORUTA	96,49	100	90,70
	RELİEF+RFE	93,86	97,18	88,37
	RELİEF +GA	93,86	98,59	86,05
	RELİEF +SA	93,86	97,18	88,37
	RELİEF +BORUTA	94,74	98,59	88,37
	SU+RFE	95,61	98,59	90,70
	SU+GA	94,74	98,59	88,37

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	SU+SA	95,61	98,59	90,70
	SU+BORUTA	95,61	98,59	90,70
J48	No Feature Selection	91,23	94,37	86,05
	IG+RFE	93,86	97,18	88,37
	IG+GA	92,11	97,18	83,72
	IG+SA	92,98	95,77	88,37
	IG+BORUTA	93,86	97,18	88,37
	GR+RFE	93,86	97,18	88,37
	GR+GA	93,86	97,18	88,37
	<b>GR+SA</b>	<b>95,61</b>	<b>100</b>	<b>88,37</b>
	GR+BORUTA	92,98	97,18	86,05
	CFS+RFE	92,11	94,37	88,37
	CFS +GA	94,74	97,18	90,70
	CFS +SA	91,23	97,18	81,40
	CFS+BORUTA	92,98	94,37	90,70
	Kİ-KARE+RFE	91,23	94,37	86,05
	Kİ-KARE +GA	94,74	98,59	88,37
	Kİ-KARE +SA	91,23	92,96	88,37
	Kİ-KARE +BORUTA	91,23	94,37	86,05
	RELİEF+RFE	91,23	94,37	86,05
	RELİEF +GA	93,86	97,18	88,37
	RELİEF +SA	92,11	97,18	83,72
	RELİEF +BORUTA	91,23	94,37	86,05
	SU+RFE	91,23	94,37	86,05
	SU+GA	93,86	98,59	86,05
	SU+SA	93,86	98,59	86,05
SU+BORUTA	92,98	97,18	86,05	
EARTH	No Feature Selection	95,61	98,59	90,70
	IG+RFE	94,74	98,59	88,37
	IG+GA	94,74	98,59	88,37
	IG+SA	92,98	95,77	88,37
	IG+BORUTA	93,86	98,59	86,05
	GR+RFE	95,61	100	88,37
	GR+GA	92,98	98,59	83,72
	GR+SA	95,61	100	88,37
	<b>GR+BORUTA</b>	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	CFS+RFE	95,61	100	88,37
	CFS +GA	95,61	98,59	90,70
	CFS +SA	92,11	98,59	81,40
	CFS+BORUTA	95,61	100	88,37
	Kİ-KARE+RFE	95,61	98,59	90,70
	Kİ-KARE +GA	94,74	98,59	88,37
	Kİ-KARE +SA	92,11	94,37	88,37
	Kİ-KARE +BORUTA	95,61	98,59	90,70
	RELİEF+RFE	95,61	100	88,37
	RELİEF +GA	94,74	98,59	88,37
	RELİEF +SA	92,98	98,59	83,72
	RELİEF +BORUTA	92,98	97,18	86,05
	SU+RFE	93,86	100	83,72
	SU+GA	92,98	94,37	90,70
	SU+SA	93,86	98,59	86,05
SU+BORUTA	93,86	100	83,72	
RF	No Feature Selection	<b>96,49</b>	<b>100</b>	<b>90,70</b>

ALGORİTMA	ÖZNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	IG+RFE	94,74	98,59	88,37
	IG+GA	94,74	97,18	90,70
	IG+SA	92,98	97,18	86,05
	IG+BORUTA	95,61	98,59	90,70
	GR+RFE	94,74	98,59	88,37
	GR+GA	95,61	100	88,37
	<b>GR+SA</b>	<b>95,61</b>	<b>100</b>	<b>88,37</b>
	GR+BORUTA	95,61	100	88,37
	CFS+RFE	94,74	98,59	88,37
	CFS +GA	95,61	98,59	90,70
	CFS +SA	92,11	97,18	83,72
	CFS+BORUTA	94,74	98,59	88,37
	Kİ-KARE+RFE	95,61	100	88,37
	Kİ-KARE +GA	94,74	100	86,05
	Kİ-KARE +SA	91,23	91,55	90,70
	Kİ-KARE +BORUTA	95,61	100	88,37
	RELİEF+RFE	95,61	100	88,37
	RELİEF +GA	93,86	97,18	88,37
	RELİEF +SA	94,74	98,59	88,37
	RELİEF +BORUTA	95,61	100	88,37
	SU+RFE	95,61	100	88,37
	SU+GA	94,74	97,18	90,70
	SU+SA	95,61	100	88,37
	SU+BORUTA	95,61	100	88,37
GBM	No Feature Selection	95,61	98,59	90,70
	IG+RFE	94,74	98,59	88,37
	IG+GA	94,74	98,59	88,37
	IG+SA	93,86	98,59	86,05
	IG+BORUTA	95,61	98,59	90,70
	GR+RFE	95,61	98,59	90,70
	GR+GA	93,86	97,18	88,37
	GR+SA	93,86	98,59	86,05
	GR+BORUTA	95,61	100	88,37
	<b>CFS+RFE</b>	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	CFS +GA	95,61	100	88,37
	CFS +SA	93,86	97,18	88,37
	CFS+BORUTA	94,74	98,59	88,37
	Kİ-KARE+RFE	95,61	98,59	90,70
	Kİ-KARE +GA	94,74	97,18	90,70
	Kİ-KARE +SA	93,86	97,18	88,37
	Kİ-KARE +BORUTA	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	RELİEF+RFE	95,61	98,59	90,70
	RELİEF +GA	93,86	97,18	88,37
	RELİEF +SA	94,74	98,59	88,37
	RELİEF +BORUTA	95,61	98,59	90,70
	SU+RFE	95,61	98,59	90,70
	SU+GA	95,61	97,18	93,02
	SU+SA	<b>96,49</b>	<b>100</b>	<b>90,70</b>
SU+BORUTA	95,61	98,59	90,70	
XGBTR EE	No Feature Selection	95,61	98,59	90,70
	IG+RFE	94,74	97,18	90,70
	IG+GA	96,49	98,59	93,02
	IG+SA	93,86	98,59	86,05

ALGORİTMA	ÖZNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	IG+BORUTA	95,61	98,59	90,70
	GR+RFE	95,61	98,59	90,70
	GR+GA	94,74	100	86,05
	GR+SA	94,74	98,59	88,37
	GR+BORUTA	94,74	98,59	88,37
	CFS+RFE	94,74	100	86,05
	CFS +GA	95,61	100	88,37
	CFS +SA	92,11	94,37	88,37
	CFS+BORUTA	95,61	100	88,37
	Kİ-KARE+RFE	95,61	98,59	90,70
	Kİ-KARE +GA	94,74	97,18	90,70
	Kİ-KARE +SA	92,98	95,77	88,37
	Kİ-KARE +BORUTA	95,61	98,59	90,70
	RELİEF+RFE	95,61	98,59	90,70
	RELİEF +GA	95,61	98,59	90,70
	RELİEF +SA	92,11	97,18	83,72
	RELİEF +BORUTA	95,61	98,59	90,70
	SU+RFE	95,61	98,59	90,70
	<b>SU+GA</b>	<b>97,37</b>	<b>100</b>	<b>93,02</b>
	SU+SA	95,61	100	88,37
	SU+BORUTA	95,61	98,59	90,70
TREEBAG	No Feature Selection	95,61	98,59	90,70
	IG+RFE	93,86	97,18	88,37
	IG+GA	93,86	97,18	88,37
	IG+SA	92,98	97,18	86,05
	IG+BORUTA	94,74	97,18	90,70
	GR+RFE	94,74	97,18	90,70
	GR+GA	93,86	98,59	86,05
	GR+SA	94,74	100	86,05
	GR+BORUTA	93,86	97,18	88,37
	CFS+RFE	93,86	97,18	88,37
	CFS +GA	93,86	97,18	88,37
	CFS +SA	92,11	97,18	83,72
	CFS+BORUTA	95,61	98,59	90,70
	Kİ-KARE+RFE	95,61	98,59	90,70
	Kİ-KARE +GA	<b>97,37</b>	<b>100</b>	93,02
	Kİ-KARE +SA	92,98	97,18	86,05
	Kİ-KARE +BORUTA	93,86	95,77	90,70
	RELİEF+RFE	93,86	98,59	86,05
	RELİEF +GA	94,74	98,59	88,37
	RELİEF +SA	92,98	98,59	83,72
	RELİEF +BORUTA	92,98	97,18	86,05
SU+RFE	94,74	97,18	90,70	
<b>SU+GA</b>	<b>97,37</b>	<b>100</b>	<b>93,02</b>	
SU+SA	93,86	100	83,72	
SU+BORUTA	93,86	95,77	90,70	
ADABOOST	No Feature Selection	96,49	100	90,70
	IG+RFE	94,74	98,59	88,37
	IG+GA	93,86	97,18	88,37
	IG+SA	94,74	97,18	90,70
	IG+BORUTA	95,61	98,59	90,70
	GR+RFE	94,74	100	86,05
	GR+GA	93,86	98,59	86,05

ALGORİTMA	ÖZİNİTELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	GR+SA	95,61	100	88,37
	GR+BORUTA	94,74	100	86,05
	CFS+RFE	93,86	98,59	86,05
	CFS +GA	95,61	98,59	90,70
	CFS +SA	91,23	94,37	86,05
	CFS+BORUTA	95,61	100	88,37
	Kİ-KARE+RFE	94,74	98,59	88,37
	Kİ-KARE +GA	95,61	100	88,37
	Kİ-KARE +SA	92,98	95,77	88,37
	Kİ-KARE +BORUTA	95,61	100	88,37
	RELİEF+RFE	95,61	100	88,37
	RELİEF +GA	95,61	100	88,37
	RELİEF +SA	94,74	98,59	88,37
	RELİEF +BORUTA	<b>96,49</b>	<b>98,59</b>	<b>93,02</b>
	SU+RFE	95,61	100	88,37
	<b>SU+GA</b>	<b>96,49</b>	<b>98,59</b>	<b>93,02</b>
	SU+SA	95,61	98,59	90,70
	SU+BORUTA	95,61	100	88,37
GLMBOOST	No Feature Selection	94,74	98,59	88,37
	IG+RFE	93,86	95,77	90,70
	IG+GA	93,86	95,77	90,70
	IG+SA	92,98	95,77	88,37
	IG+BORUTA	93,86	95,77	90,70
	GR+RFE	94,74	98,59	88,37
	GR+GA	94,74	98,59	88,37
	<b>GR+SA</b>	<b>94,74</b>	<b>100</b>	<b>86,05</b>
	GR+BORUTA	94,74	98,59	88,37
	CFS+RFE	93,86	97,18	88,37
	CFS +GA	94,74	98,59	88,37
	CFS +SA	91,23	98,59	79,07
	CFS+BORUTA	93,86	97,18	88,37
	Kİ-KARE+RFE	92,98	97,18	86,05
	Kİ-KARE +GA	93,86	98,59	86,05
	Kİ-KARE +SA	92,11	95,77	86,05
	Kİ-KARE +BORUTA	94,74	98,59	88,37
	RELİEF+RFE	92,98	97,18	86,05
	RELİEF +GA	94,74	98,59	88,37
	RELİEF +SA	91,23	98,59	79,07
	RELİEF +BORUTA	92,98	97,18	86,05
	SU+RFE	92,98	97,18	86,05
SU+GA	92,11	98,59	81,40	
SU+SA	92,98	100	88,37	
SU+BORUTA	92,98	97,18	86,05	
LOGITBOOST	No Feature Selection	94,74	98,59	88,37
	IG+RFE	92,98	97,18	86,05
	IG+GA	96,49	98,59	93,02
	IG+SA	93,86	98,59	86,05
	IG+BORUTA	95,61	97,18	93,02
	GR+RFE	96,49	98,59	93,02
	GR+GA	93,86	98,59	86,05
	GR+SA	94,74	100	86,05
	GR+BORUTA	94,74	98,59	88,37
	CFS+RFE	94,74	100	86,05

ALGORİTMA	ÖZNETELİK SEÇİM YÖNTEMİ	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
	<b>CFS +GA</b>	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	CFS +SA	93,86	97,18	88,37
	CFS+BORUTA	96,49	100	90,70
	Kİ-KARE+RFE	95,61	98,59	90,70
	Kİ-KARE +GA	95,61	97,18	93,02
	Kİ-KARE +SA	93,86	95,77	90,70
	Kİ-KARE +BORUTA	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	RELİEF+RFE	95,61	98,59	90,70
	RELİEF +GA	94,74	97,18	90,70
	RELİEF +SA	91,23	95,77	83,72
	RELİEF +BORUTA	<b>96,49</b>	<b>100</b>	<b>90,70</b>
	SU+RFE	95,61	100	88,37
	SU+GA	95,61	98,59	90,70
	SU+SA	95,61	100	88,37
	SU+BORUTA	95,61	100	88,37
ROTATION FOREST	No Feature Selection	<b>96,49</b>	<b>100</b>	<b>92,39</b>
	IG+RFE	92,98	98,59	83,72
	IG+GA	95,61	98,59	90,70
	IG+SA	93,86	97,18	88,37
	IG+BORUTA	93,86	98,59	86,05
	GR+RFE	95,61	98,59	90,70
	GR+GA	94,74	98,59	88,37
	GR+SA	94,74	98,59	88,37
	GR+BORUTA	94,74	98,59	88,37
	CFS+RFE	95,61	100	88,37
	CFS +GA	95,61	100	88,37
	CFS +SA	89,47	97,18	76,74
	CFS+BORUTA	95,61	100	88,37
	<b>Kİ-KARE+RFE</b>	<b>96,49</b>	<b>98,59</b>	<b>93,02</b>
	Kİ-KARE +GA	93,86	98,59	86,05
	Kİ-KARE +SA	90,35	92,96	86,05
	Kİ-KARE +BORUTA	94,74	98,59	88,37
	RELİEF+RFE	93,86	98,59	86,05
	RELİEF +GA	93,86	95,77	90,70
	RELİEF +SA	94,74	98,59	88,37
	RELİEF +BORUTA	92,98	97,18	86,05
	SU+RFE	92,11	97,18	83,72
	SU+GA	94,74	98,59	88,37
SU+SA	95,61	100	88,37	
SU+BORUTA	92,98	97,18	86,05	

Kaynak: Yazar tarafından oluşturulmuştur.

Yapılan çalışmada, Bölüm 3.3.2 'de de değindiğimiz farklı temel model ve meta modeller kullanılarak 15504 model denenmiştir. Bu modellerin her birinin (15504), ardışık öznetelik seçimi (24 seçim yöntemi) sonrası performansları incelenmiştir. Yani toplamda 387.600 (15504\*24) ardışık öznetelik seçim sonrası yığın modeli denenmiştir. Denenen modellere ilişkin sonuçların tamamına <https://tinyurl.com/3px2wmff> linkinden erişilebilmektedir. Burada sadece belirli

değerin üzerinde performans gösteren modeller verilmiştir. Tablo 19'da WDBC veri seti için %98 ve üzeri doğruluk değerine sahip modeller verilmiştir.

Tablo 19'da kullanılan ardışık öznitelik seçim yöntemi, kullanılan temel model, meta model ve performans ölçütlerinin değerleri (Doğruluk, Duyarlılık ve Belirleyicilik) verilmiştir. %98 ve üzeri doğruluk değerine sahip modellerin hepsi aynı Duyarlılık ve Belirleyicilik değerine sahip olduğundan; en iyi sonucu veren algoritmaya karar verebilmek için hesaplama zamanını dikkate aldığımızda SU+BORUTA ardışık öznitelik seçimi sonrası yığın algoritması (Temel model: Knn, C5.0, lda; meta model: rotationForest) vermiştir.

**Tablo 19:** WDBC Veri Seti için Ardışık Öznitelik Seçimi Sonrası %98 Üzeri Doğruluk Veren Yığın Algoritması Sonuçları

ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)	Zaman
GR+RFE	Model 319	adaboost	98,25	100	95,35	8 m
		treebag	98,25	100	95,35	4,4 m
		rf	98,25	100	95,35	4 m
GR+BORUTA	Model 90	rotationForest	98,25	100	95,35	5 m
	Model 91	LogitBoost	98,25	100	95,35	41,2 s
	Model 226	treebag	98,25	100	95,35	1,1 m
	Model 260	LogitBoost	98,25	100	95,35	47,3 s
	Model 304	rotationForest	98,25	100	95,35	5,1 m
	Model 315	rotationForest	98,25	100	95,35	5,3 m
	Model 398	naive_bayes	98,25	100	95,35	1,1 m
	Model 400	naive_bayes	98,25	100	95,35	1,1 m
	Model 407	adaboost	98,25	100	95,35	5 m
	Model 694	naive_bayes	98,25	100	95,35	5 m
Kİ-KARE+RFE	Model 119	naive_bayes	98,25	100	95,35	6 m
	Model 134	naive_bayes	98,25	100	95,35	10 m
	Model 241	glmboost	98,25	100	95,35	5,5 m
		naive_bayes	98,25	100	95,35	5,3 m
	Model 694	knn	98,25	100	95,35	9,2 m
	Model 795	treebag	98,25	100	95,35	5,5 m
		rf	98,25	100	95,35	6,1 m
	Model 810	naive_bayes	98,25	100	95,35	9,5 m
Model 811	naive_bayes	98,25	100	95,35	5 m	
Kİ-KARE+GA	Model 7	earth	98,25	100	95,35	26,2 m
	Model 150	adaboost	98,25	100	95,35	36 m
		xgbtree	98,25	100	95,35	37,5 m
		naive_bayes	98,25	100	95,35	31 m
	Model 156	treebag	98,25	100	95,35	26,2 m
	Model 210	naive_bayes	98,25	100	95,35	30,5 m
Model 384	rotationForest	98,25	100	95,35	26,3 m	
Kİ-KARE+BORUTA	Model 14	glmboost	98,25	100	95,35	1,1 m
		lda	98,25	100	95,35	1 m
	Model 29	glmboost	98,25	100	95,35	1,1 m
		rpart	98,25	100	95,35	1 m

		lda	98,25	100	95,35	1 m
		mlp	98,25	100	95,35	1,4 m
		naive_bayes	98,25	100	95,35	1 m
		rotationForest	98,25	100	95,35	1,3 m
		svmLinear	98,25	100	95,35	1 m
		svmPoly	98,25	100	95,35	3,5 m
		glm	98,25	100	95,35	1 m
	Model 43	glmboost	98,25	100	95,35	1,2 m
		lda	98,25	100	95,35	1 m
	Model 68	lda	98,25	100	95,35	1,3 m
	Model 92	treebag	98,25	100	95,35	2 m
	Model 98	glmboost	98,25	100	95,35	1,5 m
		knn	98,25	100	95,35	1,2 m
		lda	98,25	100	95,35	1,3 m
		naive_bayes	98,25	100	95,35	1,1 m
		rotationForest	98,25	100	95,35	1,3 m
		svmLinear	98,25	100	95,35	1,3 m
		svmRadial	98,25	100	95,35	1,5 m
	Model 113	glmboost	98,25	100	95,35	1,4 m
		J48	98,25	100	95,35	2,3 m
		lda	98,25	100	95,35	1,2 m
	Model 119	glmboost	98,25	100	95,35	3,4 m
		lda	98,25	100	95,35	2,4 m
		naive_bayes	98,25	100	95,35	2,4 m
	Model 134	glmboost	98,25	100	95,35	7,4 m
		lda	98,25	100	95,35	8,5 m
	Model 149	glmboost	98,25	100	95,35	1 m
		lda	98,25	100	95,35	1 m
		earth	98,25	100	95,35	1,2 m
		svmLinear	98,25	100	95,35	1,1 m
		svmPoly	98,25	100	95,35	3 m
		glm	98,25	100	95,35	1 m
	Model 163	glmboost	98,25	100	95,35	1,2 m
		lda	98,25	100	95,35	1,1 m
		svmLinear	98,25	100	95,35	1,2 m
	Model 188	glmboost	98,25	100	95,35	2 m
		lda	98,25	100	95,35	1,3 m
		mlp	98,25	100	95,35	2 m
		svmLinear	98,25	100	95,35	1,4 m
	Model 209	lda	98,25	100	95,35	56 s
	Model 218	glmboost	98,25	100	95,35	1,3 m
		knn	98,25	100	95,35	1,3 m
		lda	98,25	100	95,35	1,3 m
		earth	98,25	100	95,35	1,3 m
		naive_bayes	98,25	100	95,35	1,3 m
		svmLinear	98,25	100	95,35	1,2 m
		svmPoly	98,25	100	95,35	3,4 m
		glm	98,25	100	95,35	1,3 m
	Model 233	glmboost	98,25	100	95,35	1,4 m
		lda	98,25	100	95,35	1,4 m
	Model 239	glmboost	98,25	100	95,35	4 m
		rpart	98,25	100	95,35	2,4 m
		lda	98,25	100	95,35	3,4 m
		earth	98,25	100	95,35	3,3 m
		svmLinear	98,25	100	95,35	3,3 m

		svmPoly	98,25	100	95,35	4,4 m
		glm	98,25	100	95,35	2,4 m
	Model 254	glmboost	98,25	100	95,35	10,2 m
		knn	98,25	100	95,35	7,4 m
		lda	98,25	100	95,35	9,1 m
		mlp	98,25	100	95,35	6 m
		svmLinear	98,25	100	95,35	5,3 m
		svmPoly	98,25	100	95,35	6,2 m
		glm	98,25	100	95,35	5,1 m
	Model 255	glmboost	98,25	100	95,35	1 m
		lda	98,25	100	95,35	55 s
		svmLinear	98,25	100	95,35	56 s
	Model 268	glmboost	98,25	100	95,35	1 m
		lda	98,25	100	95,35	59 s
		svmLinear	98,25	100	95,35	1 m
	Model 293	C5.0	98,25	100	95,35	1,2 m
		lda	98,25	100	95,35	1,2 m
		rotationForest	98,25	100	95,35	1,3 m
		svmLinear	98,25	100	95,35	1,1 m
	Model 304	rotationForest	98,25	100	95,35	6,5 m
	Model 323	glmboost	98,25	100	95,35	1,2 m
		lda	98,25	100	95,35	1,1 m
		earth	98,25	100	95,35	1,2 m
		rotationForest	98,25	100	95,35	1,2 m
		svmLinear	98,25	100	95,35	1,2 m
		svmPoly	98,25	100	95,35	3 m
	Model 338	glm	98,25	100	95,35	1,1 m
		glmboost	98,25	100	95,35	1,1 m
		knn	98,25	100	95,35	1,1 m
		lda	98,25	100	95,35	1,1 m
	Model 344	rotationForest	98,25	100	95,35	1,2 m
		glmboost	98,25	100	95,35	2,2 m
		lda	98,25	100	95,35	2,1 m
		rotationForest	98,25	100	95,35	2,3 m
	Model 359	svmLinear	98,25	100	95,35	2,2 m
		glmboost	98,25	100	95,35	5,2 m
		rpart	98,25	100	95,35	5,1 m
		knn	98,25	100	95,35	5,1 m
		lda	98,25	100	95,35	5,1 m
		earth	98,25	100	95,35	5,1 m
		naive_bayes	98,25	100	95,35	5,1 m
		svmLinear	98,25	100	95,35	5,1 m
	svmPoly	98,25	100	95,35	6,3 m	
	glm	98,25	100	95,35	5,2 m	
	Model 414	lda	98,25	100	95,35	1,3 m
	Model 429	lda	98,25	100	95,35	1,2 m
	Model 435	glmboost	98,25	100	95,35	2,3 m
		lda	98,25	100	95,35	2,1 m
		rotationForest	98,25	100	95,35	2,4 m
	Model 450	glmboost	98,25	100	95,35	5,3 m
		lda	98,25	100	95,35	5,2 m
	Model 558	rpart	98,25	100	95,35	1,3 m
		lda	98,25	100	95,35	1,3 m
		naive_bayes	98,25	100	95,35	1,3 m
		svmLinear	98,25	100	95,35	1,3 m

Model 579	lda	98,25	100	95,35	2,4 m
	earth	98,25	100	95,35	2,5 m
	svmLinear	98,25	100	95,35	3 m
Model 594	J48	98,25	100	95,35	7,1 m
	C5.0	98,25	100	95,35	6,5 m
	rpart	98,25	100	95,35	6,5 m
	lda	98,25	100	95,35	6,1 m
	svmLinear	98,25	100	95,35	6,1 m
Model 595	J48	98,25	100	95,35	1,3 m
	C5.0	98,25	100	95,35	1,2 m
	rpart	98,25	100	95,35	1,2 m
Model 694	lda	98,25	100	95,35	5,4 m
	svmLinear	98,25	100	95,35	6 m
Model 709	LogitBoost	98,25	100	95,35	1,4 m
	lda	98,25	100	95,35	1,4 m
Model 715	glmboost	98,25	100	95,35	2,5 m
	J48	98,25	100	95,35	3 m
	rpart	98,25	100	95,35	2,5 m
	lda	98,25	100	95,35	3 m
	naive_bayes	98,25	100	95,35	3 m
	rotationForest	98,25	100	95,35	3,1 m
	svmLinear	98,25	100	95,35	2,5 m
	svmPoly	98,25	100	95,35	4,1 m
	glm	98,25	100	95,35	3,1 m
Model 730	glmboost	98,25	100	95,35	8,5 m
	J48	98,25	100	95,35	10,4 m
	C5.0	98,25	100	95,35	11,1 m
	rpart	98,25	100	95,35	9,4 m
	lda	98,25	100	95,35	8 m
	earth	98,25	100	95,35	10,3 m
	naive_bayes	98,25	100	95,35	7,2 m
	svmLinear	98,25	100	95,35	8,2 m
	svmPoly	98,25	100	95,35	10,2 m
	glm	98,25	100	95,35	9,5 m
Model 731	glmboost	98,25	100	95,35	1,4 m
	J48	98,25	100	95,35	1,3 m
	rpart	98,25	100	95,35	1,2 m
	lda	98,25	100	95,35	1,2 m
	earth	98,25	100	95,35	1,3 m
	svmLinear	98,25	100	95,35	1,2 m
Model 784	rotationForest	98,25	100	95,35	3 m
Model 791	rotationForest	98,25	100	95,35	4 m
Model 794	glmboost	98,25	100	95,35	8 m
	lda	98,25	100	95,35	7,5 m
	earth	98,25	100	95,35	8 m
	svmLinear	98,25	100	95,35	8 m
	svmPoly	98,25	100	95,35	9 m
	glm	98,25	100	95,35	8 m
Model 795	rpart	98,25	100	95,35	2,5 m
	lda	98,25	100	95,35	2,5 m
	earth	98,25	100	95,35	2,6 m
Model 816	glmboost	98,25	100	95,35	6,4 m
	C5.0	98,25	100	95,35	6,5 m
	lda	98,25	100	95,35	6,5 m
	earth	98,25	100	95,35	6,5 m

		svmLinear	98,25	100	95,35	6,5 m
		glm	98,25	100	95,35	6,4 m
RELİEF+BORUTA	Model 82	xgbTree	98,25	100	95,35	15,5 m
	Model 84	earth	98,25	100	95,35	1,3 m
	Model 85	adaboost	98,25	100	95,35	9,4 m
		treebag	98,25	100	95,35	3,1 m
		xgbTree	98,25	100	95,35	14,2 m
		earth	98,25	100	95,35	2,4 m
	Model 91	C5.0	98,25	100	95,35	1,2 m
		xgbTree	98,25	100	95,35	14 m
	Model 542	adaboost	98,25	100	95,35	9,5 m
		mlp	98,25	100	95,35	3,1 m
earth		98,25	100	95,35	2,1 m	
Model 561	treebag	98,25	100	95,35	2,4 m	
SU+RFE	Model 81	xgbTree	98,25	100	95,35	26,5 m
	Model 623	C5.0	98,25	100	95,35	15,1 m
	Model 658	naive_bayes	98,25	100	95,35	3,1 m
	Model 694	LogitBoost	98,25	100	95,35	10,3 m
	Model 709	naive_bayes	98,25	100	95,35	3,5 m
SU+GA	Model 39	C5.0	98,25	100	95,35	13,5 m
		rf	98,25	100	95,35	14 m
	Model 150	adaboost	98,25	100	95,35	31,1 m
		LogitBoost	98,25	100	95,35	18,2 m
		J48	98,25	100	95,35	17,2 m
		rpart	98,25	100	95,35	17,1 m
		xgbTree	98,25	100	95,35	24,2 m
	rf	98,25	100	95,35	17,1 m	
Model 164	LogitBoost	98,25	100	95,35	17,4 m	
SU+BORUTA	Model 60	rotationForest	98,25	100	95,35	35,4 s

Kaynak: Yazar tarafından oluşturulmuştur.

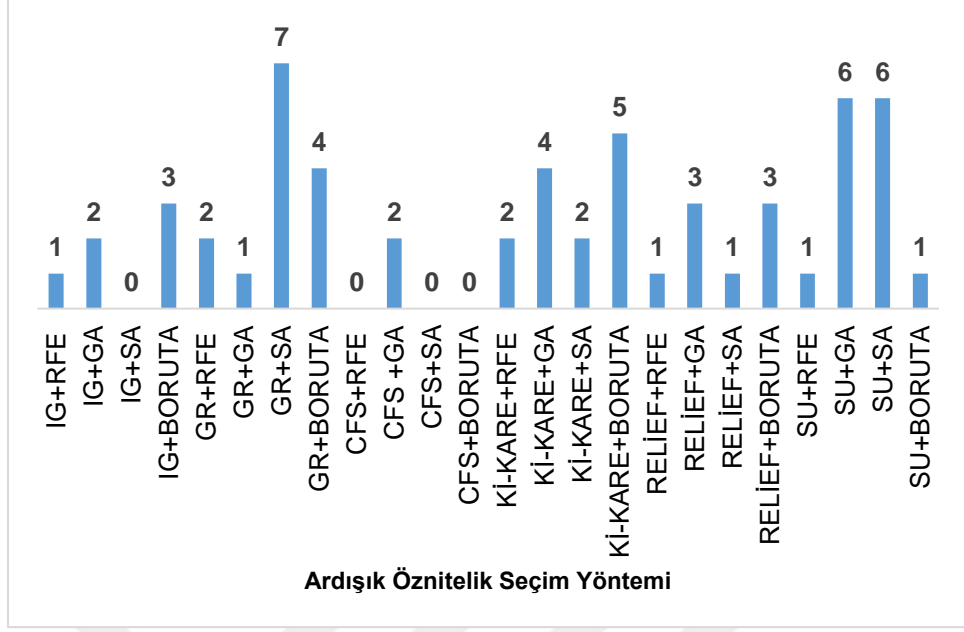
### 3.7. SONUÇ ve DEĞERLENDİRME

Yapılan çalışmada, R programında “TH.data” paketinde bulunan “GlaucomaM” veri seti ve UCI Machine Learning Repository veri tabanındaki popüler veri setlerinden “Breast Cancer Wisconsin (Diagnostic)” adlı veri seti kullanılarak algoritmalarının performansını iyileştirmek için iki farklı yaklaşım denenmiştir. İlk yaklaşım öznitelik seçimi yapmaktır. Algoritmaların performansı etkileyen en önemli unsur elimizdeki veri setindeki özniteliklerin yeterli sayıda ve gerekli olmasıdır. Bunun için sınıflandırma işlemini gerçekleştirmeden önce öznitelik seçimi yapılmıştır. Öznitelik seçimi olarak da filtreleme ve sarmal yöntemleri birleştiren hibrit bir yaklaşım kullanılmıştır. Ardışık öznitelik seçimi için filtreleme adımında Ki-Kare, Bilgi Kazancı, Kazanç Oranı, Simetrik Belirsizlik Katsayısı, Korelasyon Tabanlı Öznitelik Seçimi ve RELİEF yöntemleri kullanılırken; sarmal yöntem olarak Özyinelemeli Öznitelik

Eliminasyonu, Genetik Algoritma, Tavlama Benzetimi ve BORUTA kullanılmıştır. Ardışık öznitelik seçimi sonrası K En Yakın Komşu, Lojistik Regresyon, Naive Bayes, Destek Vektör Makinaları, Doğrusal Diskriminant Analizi, Karar Ağaçları, Rasgele Orman, Torbalama, Artırma, Rotasyon Ormanı ve Yığın algoritmaları kullanılarak sınıflandırma işlemi yapılmış ve model başarımları ölçütleri kullanılarak performansları kıyaslanmıştır. Algoritmanın performansını geliştirmek için diğer bir denenmiş yaklaşım ise, yığın algoritmasında temel ve meta model olarak farklı algoritmalar denenerak daha iyi performansa sahip model bileşimi bulunmaya çalışılmıştır. Yığın algoritmasında kullanılan temel ve meta modellere göre algoritmanın performansı değişmektedir. Temel ve meta model olarak en iyi performansı verebilecek algoritmalara ilişkin genel geçer bir kabul bulunmamaktadır. Ancak, yığın algoritmasını ortaya atan David Wolpert (1992) ikinci seviye sınıflandırıcı olarak basit doğrusal modeller veya yapraklarında doğrusal modeller bulunan ağaçların genellikle iyi çalıştığını öne sürmüştür (Witten ve diğerleri, 2005: 370-371). Ting ve Witten (1997) farklı (Temel model olarak C4.5, NB, IB1; meta model olarak C4.5, NB, IB1, MLR) 2. seviye modelleri (meta öğrenici) deneysel olarak karşılaştırmış ve basit doğrusal modelin en iyi performansı gösterdiğini bulmuşlardır (Witten ve diğerleri, 2005: 372). Yapılan çalışmada literatürden farklı olarak temel ve meta model olarak topluluk yöntemleri de denenerak kapsamlı bir model denemesi yapılmıştır ve performansları incelenmiştir.

Ardışık öznitelik seçimine ilişkin sonuçları incelediğimizde; GlaucomaM ve WDBC veri setinde de temel algoritmalara ilişkin sonuçlar incelendiğinde, her iki veri setinde de ardışık öznitelik seçimi sonrası herhangi bir öznitelik seçiminin yapılmadığı duruma (yani tüm öznitelikler kullanıldığı durum) nazaran sınıflandırma algoritmalarının performansı artmıştır. GlaucomaM veri setinde %95'in üzerinde, WDBC veri setinde ise %98'in üzerinde doğruluk değeri veren ardışık öznitelik seçim algoritmalarının frekansı Şekil 33'te verilmiştir. Şekil 33 incelendiğinde, GR+SA, SU+SA ve SU+GA yöntemleri en çok tekrar eden ardışık öznitelik seçim yöntemleridir.

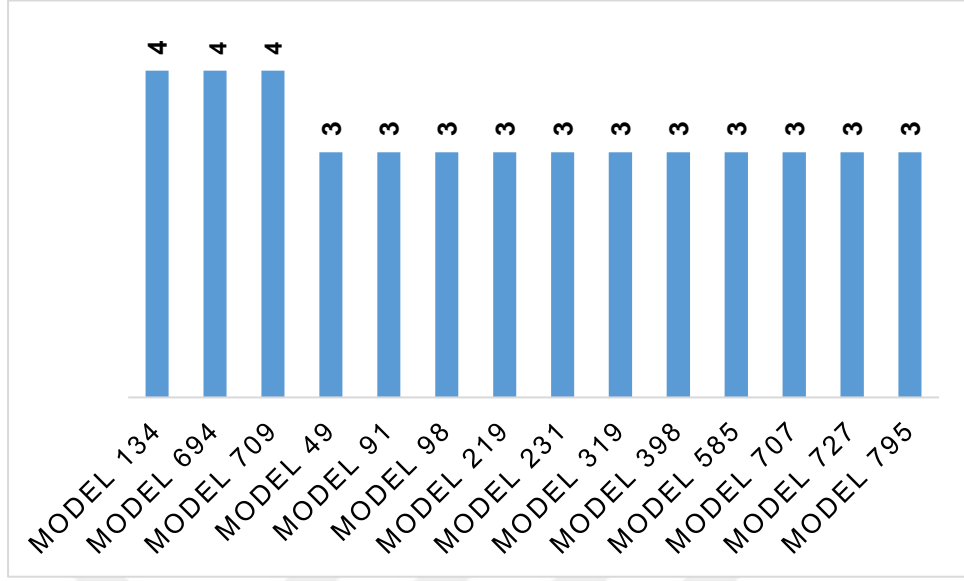
**Şekil 33:** Ardışık Öznitelik Seçim Yöntemlerinin Seçilme Sıklığı



Kaynak: Yazar tarafından oluşturulmuştur.

Yığın algoritması iki seviyeli bir sınıflandırma algoritmasıdır. Birinci seviyede ikiden fazla temel öğrenici eğitilir, daha sonra birinci seviye öğrencilerin tahminleri ikinci seviye olan meta seviye sınıflandırıcı tarafından birleştirilmektedir. Şekil 34'te, GlaucomaM veri setinde %95'in üzerinde, WDBC veri setinde ise %98'in üzerinde doğruluk değeri veren üç defa ve üzeri seçilen temel modeller verilmiştir. Seçilen temel modellerdeki algoritmalar Tablo 20'de verilmiştir. Tablo 20 incelendiğinde, topluluk öğrenmesi yöntemleri sıklıkla temel modellerde yer almaktadır. Şekil 35'te, GlaucomaM veri setinde %95'in üzerinde, WDBC veri setinde ise %98'in üzerinde doğruluk değeri veren Meta model olarak seçilmiş olan algoritmalar verilmiştir. Şekil incelendiğinde, rpart en çok tekrar ettiği görülmektedir. Meta model olarak topluluk öğrenmesi yöntemleri de sıklıkla iyi performans göstermiştir. Yığın algoritmasına ilişkin tüm sonuçları incelediğimizde; GlaucomaM ve WDBC veri setine ilişkin tüm algoritmaların sonuçları incelendiğinde, yığın algoritması en iyi sonucu vermiştir. En iyi sonucu veren algoritmalar Tablo 21'de verilmiştir.

**Şekil 34:** Üç defa ve Üzeri Seçilen Temel Modeller



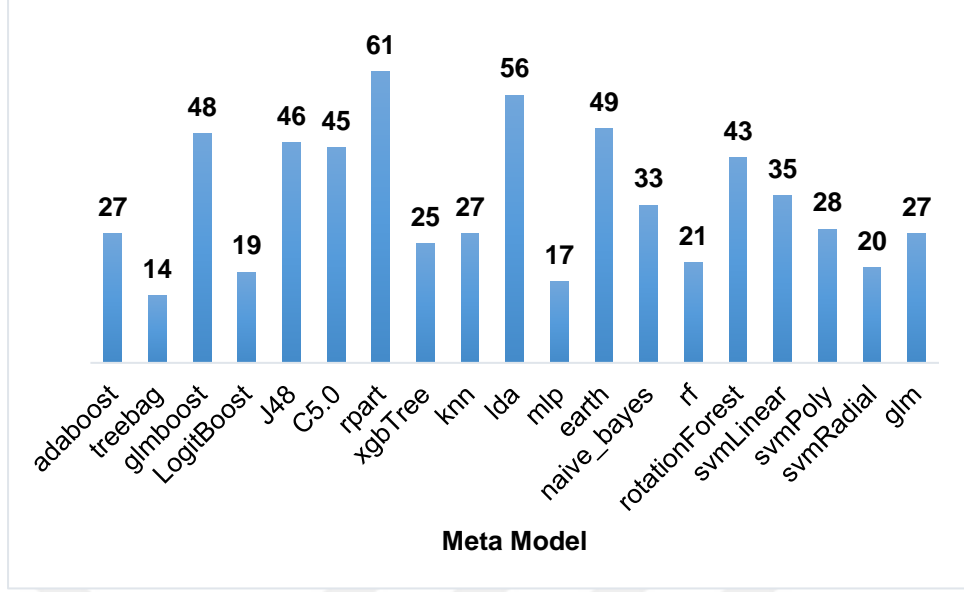
Kaynak: Yazar tarafından oluşturulmuştur.

**Tablo 20:** Üç defa ve Üzeri Seçilen Temel Modeller

Temel Model	Algoritmalar
Model 134	Knn, svmRadial, evtree
Model 694	LDA, svmRadial, evtree
Model 709	MLP, rf, svmRadial
Model 49	Knn, J48, mlp
Model 91	Knn, lda, glm
Model 98	Knn, mlp, svmRadial
Model 219	Nb, mlp, evtree
Model 231	Nb, rf, svmLinear
Model 319	Rpart, mlp, rotationForest
Model 398	Treebag, lda, mlp
Model 585	C5.0, gbm, evtree
Model 707	MLP, rf, svmLinear
Model 727	MLP, svmPoly, svmRadial
Model 795	rotationForest, svmRadial, glm

Kaynak: Yazar tarafından oluşturulmuştur.

**Şekil 35:** Meta Modellerin Seçilme Sıklığı



Kaynak: Yazar tarafından oluşturulmuştur.

**Tablo 21:** Veri Setleri için En İyi Sonucu Veren Algoritmalar

VERİ SETİ	ÖZİNİTELİK SEÇİM YÖNTEMİ	TEMEL MODEL	META MODEL	Doğruluk (%)	Duyarlılık (%)	Belirleyicilik (%)
GlaucomaM	GR+SA	Model 388	treebag	97,5	100	94,74
WDBC	SU+BORUTA	Model 60	rotationForest	98,2	100	95,35

Kaynak: Yazar tarafından oluşturulmuştur.

## SONUÇ

Yapılan çalışmada, R programında “TH.data” paketinde bulunan “GlaucomaM” veri seti ve UCI Machine Learning Repository veri tabanındaki popüler veri setlerinden “Breast Cancer Wisconsin (Diagnostic)” adlı veri seti kullanılarak algoritmalarının performansını iyileştirmek için iki farklı yaklaşım denenmiştir. İlk yaklaşım öznelik seçimi yapmaktır. Öznelik seçimi olarak da filtreleme ve sarmal yöntemleri birleştiren hibrit bir yaklaşım kullanılmıştır. Ardışık öznelik seçimine ilişkin sonuçları incelediğimizde; GlaucomaM ve WDBC veri setinde de ardışık öznelik seçimi sonrası herhangi bir öznelik seçiminin yapılmadığı duruma (yani tüm öznelikler kullanıldığı durum) nazaran sınıflandırma algoritmalarının performansı artmıştır. GlaucomaM veri setinde %95’in üzerinde, WDBC veri setinde ise %98’in üzerinde doğruluk değeri veren GR+SA, SU+SA ve SU+GA yöntemleri en çok tekrar eden ardışık öznelik seçim yöntemleridir.

Ardışık öznelik seçimi sonrası K En Yakın Komşu, Lojistik Regresyon, Naive Bayes, Destek Vektör Makinaları, Doğrusal Diskriminant Analizi, Karar Ağaçları, Rasgele Orman, Torbalama, Artırma, Rotasyon Ormanı ve Yığın algoritmaları kullanılarak sınıflandırma işlemi yapılmış ve model başarımları ölçütleri kullanılarak performansları kıyaslanmıştır.

Algoritmanın performansını geliştirmek için diğer bir denenmiş yaklaşım ise, yığın algoritmasında temel ve meta model olarak farklı algoritmalar denenerek daha iyi performansa sahip model bileşimi bulunmaya çalışılmıştır. Yığın algoritmasında kullanılan temel ve meta modellere göre algoritmanın performansı değişmektedir. Temel ve meta model olarak en iyi performansı verebilecek algoritmalara ilişkin genel geçer bir kabul bulunmamaktadır. Ancak, yığın algoritmasını ortaya atan David Wolpert (1992) ikinci seviye sınıflandırıcı olarak basit doğrusal modeller veya yapraklarında doğrusal modeller bulunan ağaçların genellikle iyi çalıştığını öne sürmüştür. Ting ve Witten (1997) farklı (Temel model olarak C4.5, NB, IB1; meta model olarak C4.5, NB, IB1, MLR) 2. seviye modelleri (meta öğrenici) deneysel olarak karşılaştırmış ve basit doğrusal modelin en iyi performansı gösterdiğini bulmuşlardır. Yapılan çalışmada literatürden farklı olarak temel ve meta model olarak topluluk yöntemleri de denenerek kapsamlı bir model denemesi yapılmıştır ve performansları incelenmiştir.

Yığın algoritması iki seviyeli bir sınıflandırma algoritmasıdır. Birinci seviyede ikiden fazla temel öğrenici eğitilir, daha sonra birinci seviye öğrenicilerin tahminleri

ikinci seviye olan meta seviye sınıflandırıcı tarafından birleştirilmektedir. GlaucomaM veri setinde %95'in üzerinde, WDBC veri setinde ise %98'in üzerinde doğruluk değeri veren yığın algoritmaları incelendiğinde; topluluk öğrenmesi yöntemlerinin sıklıkla temel modellerde yer aldığı görülmüştür. Meta model olarak GlaucomaM veri setinde %95'in üzerinde, WDBC veri setinde ise %98'in üzerinde doğruluk değeri veren algoritmalar incelendiğinde; rpart algoritmasının en çok tekrar ettiği görülmektedir. Meta model olarak topluluk öğrenmesi yöntemleri de sıklıkla iyi performans göstermiştir.

GlaucomaM ve WDBC veri setine ilişkin tüm sonuçlar incelendiğinde, ardışık öznitelik seçimi sonrası yığın algoritması en iyi sonucu verdiği görülmüştür.

Son bir değerlendirme yapmak gerekirse, hibrit öznitelik seçim yaklaşım ile sadece filtreleme yönteminin kullanıldığı duruma nazaran sınıflandırma doğruluğu iyileşmekte, aynı zamanda sadece sarmal yöntemler kullanıldığı durumdaki işlem süresi de azaltılabilmektedir. Sarmal yöntemler olarak genel olarak sezgisel algoritmaların iyi performansa sahip olduğu görülmüştür. Yığın algoritmasında farklı temel ve meta model denemeleri ile özellikle literatürdeki yaklaşımdan farklı olarak topluluk yöntemleri de kullanılarak bu algoritmaların iyi performansa sahip olduğu görülmüştür.

Gelecek çalışmalar kapsamında ele alınabilecek çalışma konuları:

- Sarmal yöntemlerde sezgisel algoritmaların iyi performansa sahip olduğu görülmüştür. Kullanılan algoritmaların (GA ve SA) parametreleri farklı metotlarla belirlenebilir ya da sarmal yöntem olarak diğer sezgisel algoritmalar denenebilir.
- Büyük veri setlerinde ardışık öznitelik seçimi yapıp hem algoritmanın performansını iyileştirilip hem de hesaplama zamanında önemli azalma meydana getirilebilir.
- Yığın algoritmasında temel model olarak üç tane model alınmıştır. Üçten fazla model denenip performansı incelenebilir.
- Çok daha farklı karakteristiğe sahip veriler kullanılarak önerilen yaklaşımlar denenerek performansları incelenebilir.

Çalışmanın kısıtlılıklarından bahsedecek olursak; İlk sınırlılığımız çok fazla model denemesi olduğundan iki veri seti üzerine deneme yapılabilmıştır ve yığın algoritmasında da üç tane temel model denenebilmiştir. Diğer sınırlılığımız ise algoritmalar sadece bir defa çalıştırılabilmıştır ve model tekrar çalıştırıldığında aynı sonucu elde edebilmek adına çekirdek (set.seed) fonksiyon kullanılmıştır.

## KAYNAKÇA

Aalaei, S., Shahraki, H., Rowhanimanesh, A., ve Eslami, S. (2016). Feature Selection Using Genetic Algorithm for Breast Cancer Diagnosis: Experiment on Three Different Datasets. *Iranian Journal of Basic Medical Sciences*. 19(5): 476.

Adler, W., Brenning, A., Potapov, S., Schmid, M., ve Lausen, B. (2011). Ensemble Classification of Paired Data. *Computational Statistics & Data Analysis*. 55(5): 1933-1941.

Adler, W., ve Lausen, B. (2009). Bootstrap Estimated True and False Positive Rates and ROC Curve. *Computational Statistics & Data Analysis*. 53(3): 718-729.

Aggarwal, C. C. (2015). *Data Mining: The Textbook*. New York: Springer.

Akay, E. Ç. (2018). Ekonometride Yeni Bir Ufuk: Büyük Veri ve Makine Öğrenmesi. *Sosyal Bilimler Araştırma Dergisi*. 7(2): 41-53.

Akay, M. F. (2009). Support Vector Machines Combined with Feature Selection for Breast Cancer Diagnosis. *Expert Systems with Applications*. 36(2): 3240-3247.

Albayrak, M., ve Albayrak, A. (2016). Mezotelyoma Hastalığı Verilerinin Sınıflandırılmasında Genetik Algoritma ile Özellik Seçimi. Paper Title (use style: paper title) (researchgate.net)

Alpaydin, E. (2020). *Introduction to Machine Learning*. Cambridge: MIT press.

Altunkaynak, B. (2019). *Veri Madenciliği Yöntemleri ve R Uygulamaları*. Ankara: Seçkin Yayıncılık.

Anagnostopoulos, I., Anagnostopoulos, C., Vergados, D., Rouskas, A., ve Kormontzas, G. (2006). The Wisconsin Breast Cancer Problem: Diagnosis and TTR/DFS Time Prognosis Using Probabilistic and Generalised Regression Information Classifiers. *Oncology Reports*. 15(4): 975-981.

Aras, S. (2013). *Yapay Sinir Ağlarıyla Zaman Serisi Tahminlemede Yeni Bir Model Seçim Stratejisi*. (Yayınlanmış Doktora Tezi). İzmir: Dokuz Eylül Üniversitesi, Sosyal Bilimler Enstitüsü, Ekonometri Anabilim Dalı.

Bagui, S. C., Bagui, S., Pal, K., ve Pal, N. R. (2003). Breast Cancer Detection Using Rank Nearest Neighbor Classification Rules. *Pattern Recognition*. 36(1): 25-34.

Bahety, A. (2014). Extension and Evaluation of ID3 – Decision Tree Algorithm. *Entropy*. 2(1): 1–8.

Balaban, M. E., ve Kartal, E. (2015). *Veri Madenciliği ve Makine Öğrenmesi*. İstanbul: Çağlayan Kitabevi.

Bell, J. (2020). *Machine Learning: Hands-On for Developers and Technical Professionals*. New York: John Wiley & Sons.

Bennett, K. P., ve Blue, J. A. (1998). A Support Vector Machine Approach to Decision Trees. In *1998 IEEE International Joint Conference on Neural Networks Proceedings: IEEE World Congress on Computational Intelligence* (ss. 2396-2401), Anchorage, Alaska, USA. 4-9 Mayıs 1998.

Bilgin, M. (2018). *Makine Öğrenmesi*. İstanbul: Papatya Yayınları.

Breiman, L. (1996). Stacked Regression. *Machine Learning*. 24(1): 49–64.

Briem, G. J., Benediktsson, J. A., ve Sveinsson, J. R. (2002). Multiple Classifiers Applied to Multisource Remote Sensing Data. *IEEE Transactions on Geoscience and Remote Sensing*. 40(10): 2291-2299.

Brijain, M., Patel, R., Kushik, M. R., ve Rana, K. (2014). A Survey on Decision Tree Algorithm For Classification. *International Journal of Engineering Development and Research*. 2(1): 1–5.

Budak, H. (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*. 22: 21-31.

Burke, L. I., ve Ignizio, J. P. (1992). Neural Networks and Operations Research: An Overview. *Computers & Operations Research*. 19(3-4): 179-189.

Cai, J., Luo, J., Wang, S., ve Yang, S. (2018). Feature Selection in Machine Learning: A New Perspective. *Neurocomputing*. 300: 70-79.

Canuto, A. M., ve Nascimento, D. S. (2012). A Genetic-Based Approach to Features Selection for Ensembles Using a Hybrid and Adaptive Fitness Function. *In The 2012 International Joint Conference On Neural Networks (IJCNN)* (ss. 1-8). Brisbane, Australia. 10-15 Haziran 2012.

Chao, W. (2011). Machine Learning Tutorial. Taiwan: National Taiwan University, Graduate Institute of Communication Engineering, DISP Lab. Machine Learning Tutorial.pdf (ntu.edu.tw)

Chaudhary, A., Kolhe, S., ve Kamal, R. (2016). An Improved Random Forest Classifier for Multi-Class Classification. *Information Processing in Agriculture*. 3(4): 215-222.

Chen, H. L., Yang, B., Liu, J., ve Liu, D. Y. (2011). A Support Vector Machine Classifier with Rough Set-Based Feature Selection for Breast Cancer Diagnosis. *Expert Systems with Applications*. 38(7): 9014-9022.

Cortes, C., ve Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*. 20(3): 273-297.

Çölkesen, İ., ve Ertekin, Ö. H. (2018). Hiperspektral Görüntülerin Sınıflandırılmasında Gelişmiş Karar Ormanı Toplu Öğrenme Algoritmalarının Performans Analizi. *VII. Uzaktan Algılama-CBS Sempozyumu (UZAL-CBS)*, Eskişehir. 18-21 Eylül 2018.

Das, S. (2001). Filters, Wrappers and A Boosting-Based Hybrid for Feature Selection. *In Icml*. 1: 74-81.

De Mántaras, R. L. (1991). A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*. 6(1): 81-92.

Deka, P. C. (2014). Support Vector Machine Applications in the Field of Hydrology: A Review. *Applied Soft Computing*. 19: 372-386.

Demirhan, T. (2015). *Makine Öğrenmesi Algoritmalarının Karmaşıklık ve Doğunluk Analizinin Bir Veri Kümesi Üzerinde Gerçekleştirilmesi*. (Yayınlanmamış Doktora Tezi). Edirne: Trakya Üniversitesi Fen Bilimleri Enstitüsü.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *In International Workshop On Multiple Classifier Systems*: 1-15. Springer, Berlin, Heidelberg.

Dora, L., Agrawal, S., Panda, R., ve Abraham, A. (2017). Optimal Breast Cancer Classification Using Gauss–Newton Representation Based Algorithm. *Expert Systems with Applications*. 85: 134-145.

El Naqa, I., ve Murphy, M. J. (2015). What is Machine Learning? *Machine Learning in Radiation Oncology* (ss. 3–11). Cham: Springer.

Elmas, Ç. (2010). *Yapay Zekâ Uygulamaları*. Ankara: Seçkin Yayıncılık.

Friedman, J., Hastie, T., ve Tibshirani, R. (2001). *The Elements of Statistical Learning*. New York: Springer.

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*. 3: 1289-1305. <https://www.jmlr.org/papers/v3/forman03a.html>.

Fraiwan, L., Lweesy, K., Khasawneh, N., Wenz, H., ve Dickhaus, H. (2012). Automated Sleep Stage Identification System Based on Time–Frequency Analysis of a Single EEG Channel and Random Forest Classifier. *Computer Methods And Programs In Biomedicine*. 108(1): 10-19.

Gislason, P. O., Benediktsson, J. A., ve Sveinsson, J. R. (2006). Random Forests for Land Cover Classification. *Pattern Recognition Letters*. 27(4): 294-300.

Gokulnath, C. B., ve Shantharajah, S. P. (2019). An Optimized Feature Selection Based on Genetic Approach and Support Vector Machine for Heart Disease. *Cluster Computing*. 22(6): 14777-14787.

Granitto, P. M., Furlanello, C., Biasioli, F., ve Gasperi, F. (2006). Recursive Feature Elimination with Random Forest for PTR-MS Analysis of Agroindustrial Products. *Chemometrics and Intelligent Laboratory Systems*. 83(2): 83-90.

Gul, A., Khan, Z., Perperoglou, A., Mahmoud, O., Miftahuddin, M., Adler, W., ve Lausen, B. (2016). *Ensemble of Subset of K-Nearest Neighbours Models for Class Membership Probability Estimation*. In *Analysis of Large and Complex Data* (ss. 411-421). Cham, Switzerland: Springer.

Gul, A., Perperoglou, A., Khan, Z., Mahmoud, O., Miftahuddin, M., Adler, W., ve Lausen, B. (2018). Ensemble of a Subset of Knn Classifiers. *Advances in Data Analysis and Classification*. 12(4): 827-840.

Guyon, I., ve Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*. 3(3): 1157-1182.

Gümüřçü, A., Aydılek, İ. B., ve Tařaltın, R. (2016). Mikro-dizilim Veri Sınıflandırmasında Öznitelik Seçme Algoritmalarının Karşılaştırılması. *Harran Üniversitesi Mühendislik Dergisi*. 1(1): 1-7.

Gündüz, H. (2019). Parkinson Hastalığı Tespitinde Farklı Boyutsallık İndirgeme Yöntemlerinin Karşılaştırılması. *Avrupa Bilim ve Teknoloji Dergisi*. (17): 1164-1172.

Gürsakar, N. (2019). *Veri Bilimi*. Bursa: Dora Yayıncılık.

Hall, M. (1999). *Correlation-Based Feature Selection for Machine Learning*. (Yayınlanmış Doktora Tezi). Department of Computer Science, Waikato University, New Zealand.

Han, H., Li, Y., ve Zhu, X. (2019). Convolutional Neural Network Learning for Generic Data Classification. *Information Sciences*. 477: 448-465

Han, J., Pei, J., ve Kamber, M. (2011). *Data Mining: Concepts and Techniques (Third Edition)*. San Francisco: Morgan Kaufmann.

Hothorn, T., ve Lausen, B. (2003). Bagging Tree Classifiers for Laser Scanning Images: A Data-and Simulation-Based Strategy. *Artificial Intelligence in Medicine*. 27(1): 65-79.

Hothorn, T., ve Lausen, B. (2003). Double-Bagging: Combining Classifiers by Bootstrap Aggregation. *Pattern Recognition*. 36(6): 1303-1309.

Hothorn, T., ve Lausen, B. (2005). Bundling Classifiers by Bagging Trees. *Computational Statistics & Data Analysis*. 49(4): 1068-1078.

Huang, C. L., ve Wang, C. J. (2006). A GA-Based Feature Selection and Parameters Optimization for Support Vector Machines. *Expert Systems with Applications*. 31(2): 231-240.

Huang, C. L., ve Dun, J. F. (2008). A Distributed PSO–SVM Hybrid System with Feature Selection and Parameter Optimization. *Applied Soft Computing*. 8(4): 1381-1391.

Hussain, M., Wajid, S. K., Elzaart, A., ve Berbar, M. (2011). A Comparison of SVM Kernel Functions for Breast Cancer Detection. *In 2011 Eight International Conference Computer Graphics, Imaging and Visualization* (ss. 145-150), Washington, DC, United States. 17-19 Ağustos 2011.

Ibrahim, R. A., Ewees, A. A., Oliva, D., Abd Elaziz, M., ve Lu, S. (2019). Improved Salp Swarm Algorithm Based on Particle Swarm Optimization for Feature Selection. *Journal of Ambient Intelligence and Humanized Computing*. 10(8): 3155-3169.

James, G., Witten, D., Hastie, T., ve Tibshirani, R. (2013). *An Introduction to Statistical Learning*. New York: Springer.

Kabacoff, R. I. (2015). *Data Analysis and Graphics with R*. Shelter Island: Manning Publications.

Kadavi, P. R., Lee, C. W., ve Lee, S. (2018). Application of Ensemble-Based Machine Learning Models to Landslide Susceptibility Mapping. *Remote Sensing*. 10(8): 1252.

Kantardzic, M. (2011). *Data Mining: Concepts, Models, Methods, and Algorithms*. New York: John Wiley & Sons Inc publishes.

Kapucu, C., ve Çubukçu, M. (2019). Fotovoltaik Sistemlerde Topluluk Öğrenmesi Temelli Hata Tespiti. *Bilişim Teknolojileri Dergisi*. 12(2): 83-91.

Kavzoğlu, T., Çölkesen, Ğ., ve Case, K. (2010). Karar Ağaçları ile Uydu Görüntülerinin Sınıflandırılması: Kocaeli Örneği. *Harita Teknolojileri Elektronik Dergisi*. 2(1): 36–45.

Karabatak, M., ve Ince, M. C. (2009). An Expert System for Detection of Breast Cancer Based on Association Rules and Neural Network. *Expert Systems with Applications*. 36(2): 3465-3469.

Kaynar, O., Arslan, H., Görmez, Y., ve Işık, Y. E. (2018). Makine Öğrenmesi ve Öznitelik Seçim Yöntemleriyle Saldırı Tespiti. *Bilişim Teknolojileri Dergisi*. 11(2): 175-185.

Ketabchi, S., Moosaei, H., Razzaghi, M., ve Pardalos, P. M. (2019). An Improvement on Parametric  $v$ -Support Vector Algorithm for Classification. *Annals of Operations Research*. 276(1): 155-168.

Khalid, S., Khalil, T., ve Nasreen, S. (2014). A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning. *In 2014 Science and Information Conference* (ss. 372-378), London, United Kingdom. 27-29 Ağustos 2014.

Khan, Z., Gul, A., Perperoglou, A., Miftahuddin, M., Mahmoud, O., Adler, W., ve Lausen, B. (2020). Ensemble Of Optimal Trees, Random Forest and Random Projection Ensemble Classification. *Advances in Data Analysis and Classification*. 14(1): 97-116.

Kira, K., ve Rendell, L. A. (1992). The Feature Selection Problem: Traditional Methods and a New Algorithm. *In Aaii*. Volume 2: 129-134.

Kiyan, T., ve Yildirim, T. (2004). Breast Cancer Diagnosis Using Statistical Neural Networks. *IU-Journal of Electrical & Electronics Engineering*. 4(2): 1149-1153.

Kocakoç, İ. D. ve Pulat, M. (2022). Sınıflandırma Karar Ağaçları. *Sosyal Bilimler için Çok Değişkenli İstatistiksel Analiz Teknikleri: STATA, SPSS ve R Uygulamaları* (ss.327-363). Ankara: Nobel Yayınevi.

Kumar, U. K., Nikhil, M. S., ve Sumangali, K. (2017). Prediction of Breast Cancer Using Voting Classifier Technique. *In 2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)* (ss. 108-114), Chennai, India. 2-4 Ağustos 2017.

Kursa, M. B., ve Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software*. 36: 1-13.

Kutlugün, M. A. (2017). *Gözetimli Makine Öğrenmesi Yoluyla Türe Göre Metinden Ses Sentezleme*. (Yayınlanmış Yüksek Lisans Tezi). İstanbul: İstanbul Sabahattin Zaim Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Latchoumi, T. P., ve Parthiban, L. (2017). Abnormality Detection Using Weighed Particle Swarm Optimization and Smooth Support Vector Machine. *Biomedical Research*. 28(11): 4749-4751.

Lavanya, D., ve Rani, D. K. U. (2011). Analysis of Feature Selection with Classification: Breast Cancer Datasets. *Indian Journal of Computer Science and Engineering (IJCSE)*. 2(5): 756-763.

Lewis, R. J., Ph, D., ve Street, W. C. (2000). An Introduction to Classification and Regression Tree (CART) Analysis. *2000 Annual Meeting of the Society for Academic Emergency Medicine* (ss. 1–14), San Francisco, California, USA. 22- 25 Mayıs 2000.

Liaw, A., ve Wiener, M. (2002). Classification and Regression by randomForest. *R News*. 2(3): 18-22.

Lin, S. W., Ying, K. C., Chen, S. C., ve Lee, Z. J. (2008). Particle Swarm Optimization for Parameter Determination And Feature Selection of Support Vector Machines.

*Expert Systems with Applications*. 35(4): 1817-1824.

Liu, X., ve Fu, H. (2012). A Hybrid Algorithm for Text Classification Problem. *Prz Electrotechniczn*. 1: 8-11.

Liu, N., Qi, E. S., Xu, M., Gao, B., ve Liu, G. Q. (2019). A Novel Intelligent Classification Model for Breast Cancer Diagnosis. *Information Processing & Management*. 56(3): 609-623.

Maldonado, S., Weber, R., ve Basak, J. (2011). Simultaneous Feature Selection and Classification Using Kernel-Penalized Support Vector Machines. *Information Sciences*. 181(1): 115-128.

Marcano-Cedeño, A., Quintanilla-Domínguez, J., ve Andina, D. (2011). WBCD Breast Cancer Database Classification Applying Artificial Meta Plasticity Neural Network. *Expert Systems with Applications*. 38(8): 9573-9579.

Medjahed, S. A., Saadi, T. A., ve Benyettou, A. (2013). Breast Cancer Diagnosis by Using K-Nearest Neighbor with Different Distances and Classification Rules. *International Journal of Computer Applications*. 62(1): 1-5.

Mitchell, T. M. (1997). *Machine Learning*. USA: MacGraw-Hil.

Mlxtend. [https://rasbt.github.io/mlxtend/user\\_guide/classifier/LogisticRegression/](https://rasbt.github.io/mlxtend/user_guide/classifier/LogisticRegression/), (15.05.2023).

Montana, D. J., ve Davis, L. (1989). Training Feedforward Neural Networks Using Genetic Algorithms. *In IJCAI*. 89: 762-767.

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., ve Brown, S. D. (2004). An Introduction to Decision Tree Modeling. *Journal of Chemometrics*. 18(6): 275–285.

Nahato, K. B., Harichandran, K. N., ve Arputharaj, K. (2015). Knowledge Mining from Clinical Datasets Using Rough Sets and Backpropagation Neural Network. *Computational and Mathematical Methods in Medicine*. 1-13.

Nazlı, R. (20 Haziran 2021). Medium, R ile Bagging Yöntemi (Bootstrap Aggregating). <https://medium.com/@rmys.nzl/r-ile-bagging-yöntemi-bootstrap-aggregating-40b89c656759>, (15.05.2023).

Nilashi, M., Ibrahim, O., Ahmadi, H., ve Shahmoradi, L. (2017). A Knowledge-Based System for Breast Cancer Classification Using Fuzzy Logic Method. *Telematics and Informatics*. 34(4): 133-144.

Nilsson, N. J. (1996). *Introduction to Machine Learning. An Early Draft of A Proposed Textbook*. Robotics Laboratory, Department of Computer Science, Stanford University: Stanford. <http://robotics.stanford.edu/people/nilsson/mlbook>.

Nishat, M. M., Hasan, T., Nasrullah, S. M., Faisal, F., Asif, M. A. A. R., ve Hoque, M. A. (2021). Detection of Parkinson's Disease by Employing Boosting Algorithms. In *2021 Joint 10th International Conference on Informatics, Electronics & Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (ss. 1-7), Virtual Conference. 16-19 Ağustos 2021.

Niuniu, X., ve Yuxun, L. (2010). Notice of Retraction: Review of Decision Trees. *Proceedings of the Third IEEE International Conference on Computer Science and Information Technology* (ss. 105–109), Chengdu, China. 9-11 Haziran 2010.

Novaković, J. (2016). Toward Optimal Feature Selection Using Ranking Methods and Classification Algorithms. *Yugoslav Journal of Operations Research*. 21(1): 119-135.

Obaid, O. I., Mohammed, M. A., Ghani, M. K. A., Mostafa, A., ve Taha, F. (2018). Evaluating the Performance of Machine Learning Techniques in the Classification of Wisconsin Breast Cancer. *International Journal of Engineering & Technology*. 7(4): 160-166.

Omondiagbe, D. A., Veeramani, S., ve Sidhu, A. S. (2019, April). Machine Learning Classification Techniques for Breast Cancer Diagnosis. In *IOP Conference Series: Materials Science and Engineering*. 495 (1): 1-16.

Onan, A. (2018). A Clustering Based Classifier Ensemble Approach to Corporate Bankruptcy Prediction. *Alphanumeric Journal*. 6(2): 365-376.

Onan, A., Korukođlu, S., ve Bulut, H. (2016). A Multiobjective Weighted Voting Ensemble Classifier Based on Differential Evolution Algorithm for Text Sentiment Classification. *Expert Systems with Applications*. 62: 1-16.

OPENGENUS. <https://iq.opengenus.org/types-of-boosting-algorithms/>, (15.05.2023).

Özkan, Y. (2016). *Veri Madenciliđi Yöntemleri*. İstanbul: Papatya Yayınları.

Öztemel, E. (2003). *Yapay Sinir Ağları*. İstanbul: Papatya Yayınları.

Pach, F. P., ve Abonyi, J. (2006). Association Rule and Decision Tree Based Methods for Fuzzy Rule Base Generation. *World Academy of Science, Engineering and Technology*. 13: 45-50.

Pandya, R., ve Pandya, J. (2015). C5.0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning. *International Journal of Computer Applications*. 117(16): 18-21.

Patel, J., Shah, S., Thakkar, P., ve Kotecha, K. (2015). Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. *Expert Systems with Applications*. 42(1): 259-268.

Patil, T. R., ve Sherekar, S. S. (2013). Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal of Computer Science and Applications*. 6(2): 256-261.

Peker, M., Arslan, A., Şen, B., Çelebi, F. V., ve But, A. (2015). A Novel Hybrid Method for Determining the Depth of Anesthesia Level: Combining ReliefF Feature Selection and Random Forest Algorithm (ReliefF+ RF). In *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)* (ss. 1-8), Madrid, Spain. 2-4 Eylül 2015.

Pena-Reyes, C. A., ve Sipper, M. (1999). A Fuzzy-Genetic Approach to Breast Cancer Diagnosis. *Artificial Intelligence in Medicine*. 17(2): 131-155.

Polat, K., ve Güneş, S. (2007). Breast Cancer Diagnosis Using Least Square Support Vector Machine. *Digital Signal Processing*. 17(4): 694-701.

Priyam, A., Gupta, R., Rathee, A., ve Srivastava, S. (2013). Comparative Analysis of Decision Tree Classification Algorithms. *International Journal of Current Engineering and Technology*. 3(2): 334–337.

Pulat, M., ve Kocakoç, İ. D. (2017). Gezgin Satıcı Probleminin Çözümünde Kullanılan Genetik Algoritmanın Parametrelerinin İncelenmesi. *Uluslararası İktisadi ve İdari İncelemeler Dergisi*. 21-36.

Pulat, M., ve Kocakoç, İ. D. (2019). Gezgin Satıcı Probleminin Genetik Algoritmalar Kullanarak Çözümünde Çaprazlama Operatörlerinin Örnek Olaylar Bazlı İncelenmesi. *İzmir İktisat Dergisi*. 34(2): 225-243.

Pulat, M., ve Kocakoç, İ. D. (2021). Türkiye’de Makine Öğrenmesi ve Karar Ağaçları Alanında Yayımlanmış Tezlerin Bibliyometrik Analizi. *Yönetim ve Ekonomi Dergisi*. 28(2): 287-308.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. USA: Morgan Kaufmann.

Quinlan, J. R. (1996). Bagging, Boosting, and C4. 5. In *Aai/laai*. 1: 725-730.

Quinlan, J. R. (1996). Improved Use of Continuous Attributes in C4. 5. *Journal of Artificial Intelligence Research*. 4: 77-90.

Rangarajan, L. (2010). Bi-Level Dimensionality Reduction Methods Using Feature Selection and Feature Extraction. *International Journal of Computer Applications*. 4(2): 33-38.

Rebala, G., Ravi, A., ve Churiwala, S. (2019). *An Introduction to Machine Learning*. Springer.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., ve Chica-Rivas, M. J. O. G. R. (2015). Machine Learning Predictive Models For Mineral Prospectivity: An

Evaluation of Neural Networks, Random Forest, Regression Trees and Support Vector Machines. *Ore Geology Reviews*. 71: 804-818.

Rodriguez, J. J., Kuncheva, L. I., ve Alonso, C. J. (2006). Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28(10): 1619-1630.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., ve Chica-Rivas, M. J. O. G. R. (2015). Machine Learning Predictive Models for Mineral Prospectivity: An Evaluation of Neural Networks, Random Forest, Regression Trees and Support Vector Machines. *Ore Geology Reviews*. 71: 804-818.

Saeyns, Y., Inza, I., ve Larrañaga, P. (2007). A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics*. 23(19): 2507-2517.

Salama, G. I., Abdelhalim, M. B., ve Zeid, M. A. E. (2012). Experimental Comparison of Classifiers for Breast Cancer Diagnosis. *In 2012 Seventh International Conference on Computer Engineering & Systems (ICCES)* (ss. 180-185). Cairo, Egypt. 27-29 Kasım 2012.

Salunkhe, U. R., ve Mali, S. N. (2016). Classifier Ensemble Design for Imbalanced Data Classification: A Hybrid Approach. *Procedia Computer Science*. 85: 725-732.

Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*. 3(3): 210-229.

Sarafrazi, S., ve Nezamabadi-Pour, H. (2013). Facing the Classification of Binary Problems with a GSA-SVM Hybrid System. *Mathematical and Computer Modelling*. 57(1-2): 270-278.

Seera, M., ve Lim, C. P. (2014). A Hybrid Intelligent System for Medical Data Classification. *Expert Systems with Applications*. 41(5): 2239-2249.

Setiono, R. (1996). Extracting Rules from Pruned Neural Networks for Breast Cancer Diagnosis. *Artificial Intelligence in Medicine*. 8(1): 37-51.

Setiono, R. (2000). Generating Concise and Accurate Classification Rules for Breast Cancer Diagnosis. *Artificial Intelligence in Medicine*, 18(3): 205-219.

Sevli, O. (2019). Göğüs Kanseri Teşhisinde Farklı Makine Öğrenmesi Tekniklerinin Performans Karşılaştırması. *Avrupa Bilim ve Teknoloji Dergisi*. (16): 176-185.

Silahtaroglu, G. (2016). *Veri Madenciliği*. İstanbul: Papatya Yayınları.

Song, Y., ve Lu, Y. (2015). Decision Tree Methods: Applications for Classification and Prediction. *Shanghai Archives of Psychiatry*. 27(2): 130–135.

Subanya, B., ve Rajalaxmi, R. R. (2014). Feature selection using Artificial Bee Colony for cardiovascular disease classification. *In 2014 International Conference on Electronics and Communication Systems (ICECS)* (ss. 1-6), Coimbatore, India. 13-14 Şubat 2014.

Sumbaly, R., Vishnusri, N., ve Jeyalatha, S. (2014). Diagnosis of Breast Cancer Using Decision Tree Data Mining Technique. *International Journal of Computer Applications*. 98(10). 16-24.

Sun, S., ve Huang, R. (2010). An Adaptive K-Nearest Neighbor Algorithm. *In 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery* (ss. 91-94), Yantai, China. 10-12 Ağustos 2010.

Tabakhi, S., Moradi, P., ve Akhlaghian, F. (2014). An Unsupervised Feature Selection Algorithm Based on Ant Colony Optimization. *Engineering Applications of Artificial Intelligence*. 32: 112-123.

TIBCO. <https://www.tibco.com/reference-center/what-is-a-random-forest>, (15.05.2023)

Ting, K. M., ve Witten, I. H. (1997). Stacked Generalization: When Does it Work? *In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (ss. 866–871), Nagoya, Japan. 23- 29 Ağustos 1997.

Tsai, C. F., Eberle, W., ve Chu, C. Y. (2013). Genetic Algorithms in Feature and Instance Selection. *Knowledge-Based Systems*. 39: 240-247.

Tutz, G., ve Koch, D. (2016). Improved Nearest Neighbor Classifiers by Weighting and Selection of Predictors. *Statistics and Computing*. 26(5): 1039-1057.

UCI Machine Learning Repository.

[https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+\(diagnostic\)](https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)),  
(15.05.2023)

Uğuz, H. (2011). A Two-Stage Feature Selection Method for Text Categorization by Using Information Gain, Principal Component Analysis and Genetic Algorithm. *Knowledge-Based Systems*. 24(7): 1024-1032.

Verma, A., ve Mehta, S. (2017). A comparative study of ensemble learning methods for classification in bioinformatics. *In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence* (ss. 155-158), Noida, India. 2-13 Ocak 2017.

Wang, Z., Wang, Y., ve Srinivasan, R. S. (2018). A Novel Ensemble Learning Approach to Support Building Energy Use Prediction. *Energy and Buildings*. 159: 109-122.

Wang, H., Zheng, B., Yoon, S. W., ve Ko, H. S. (2018). A Support Vector Machine-Based Ensemble Algorithm for Breast Cancer Diagnosis. *European Journal of Operational Research*. 267(2): 687-699.

Welikala, R. A., Fraz, M. M., Dehmeshki, J., Hoppe, A., Tah, V., Mann, S., ve Barman, S. A. (2015). Genetic Algorithm Based Feature Selection Combined with Dual Classification for the Automated Detection of Proliferative Diabetic Retinopathy. *Computerized Medical Imaging and Graphics*. 43: 64-77.

Wen, X., Shao, L., Xue, Y., ve Fang, W. (2015). A Rapid Learning Algorithm for Vehicle Classification. *Information Sciences*. 295: 395-406.

Widodo, A., ve Yang, B. S. (2007). Support Vector Machine in Machine Condition Monitoring and Fault Diagnosis. *Mechanical Systems and Signal Processing*. 21(6): 2560-2574.

Witten, I. H., Frank, E. ve Hall, M. A. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. USA: Morgan Kaufmann Publishers.

Wolberg, W. H., Street, W. N., ve Mangasarian, O. L. (1994). Machine Learning Techniques to Diagnose Breast Cancer from Image-Processed Nuclear Features of Fine Needle Aspirates. *Cancer Letters*. 77(2-3): 163-171.

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*. 5: 241–259.

Xie, J., ve Wang, C. (2011). Using Support Vector Machines with a Novel Hybrid Feature Selection Method for Diagnosis of Erythematous-Squamous Diseases. *Expert Systems with Applications*. 38(5): 5809-5815.

Yıldız O., Tez M., Bilge H.Ş., Akcayol M.A., Güler İ., Meme Kanseri Sınıflandırması için Gen Seçimi. *20nd Signal Processing and Communications Applications Conference* (ss. 988-991), Muğla, Turkey. 18-20 Nisan 2012.

Zhang, J., Chen, L., Tian, J. X., Abid, F., Yang, W., ve Tang, X. F. (2021). Breast Cancer Diagnosis Using Cluster-based Undersampling and Boosted C5. 0 Algorithm. *International Journal of Control, Automation and Systems*. 19(5): 1998-2008.

Zheng, B., Yoon, S. W., ve Lam, S. S. (2014). Breast Cancer Diagnosis Based on Feature Extraction Using a Hybrid of K-Means and Support Vector Machine Algorithms. *Expert Systems with Applications*. 41(4): 1476-1482.

Zhou, Z. H. (2012). *Ensemble Methods: Foundations and Algorithms*. New York: CRC press.

## EKLER



## EK 1: Yiğın Algoritmasında Denenmiş Temel Modeller

1. Knn, nb, rpart
2. Knn, nb, treebag
3. Knn, nb, J48
4. Knn, nb, C5.0
5. Knn, nb, xgbTree
6. Knn, nb, lda
7. Knn, nb, mlp
8. Knn, nb, earth
9. Knn, nb, rf
10. Knn, nb, rotationForest
11. Knn, nb, gbm
12. Knn, nb, svmLinear
13. Knn, nb, svmPoly
14. Knn, nb, svmRadial
15. Knn, nb, evtree
16. Knn, nb, glm
17. Knn, rpart, treebag
18. Knn, rpart, J48
19. Knn, rpart, C5.0
20. Knn, rpart, xgbTree
21. Knn, rpart, lda
22. Knn, rpart, mlp
23. Knn, rpart, earth
24. Knn, rpart, rf
25. Knn, rpart, rotationForest
26. Knn, rpart, gbm
27. Knn, rpart, svmLinear
28. Knn, rpart, svmPoly
29. Knn, rpart, svmRadial
30. Knn, rpart, evtree
31. Knn, rpart, glm
32. Knn, treebag, J48
33. Knn, treebag, C5.0
34. Knn, treebag, xgbTree
35. Knn, treebag, lda
36. Knn, treebag, mlp
37. Knn, treebag, earth
38. Knn, treebag, rf
39. Knn, treebag, rotationForest
40. Knn, treebag, gbm
41. Knn, treebag, svmLinear
42. Knn, treebag, svmPoly
43. Knn, treebag, svmRadial
44. Knn, treebag, evtree
45. Knn, treebag, glm
46. Knn, J48, C5.0
47. Knn, J48, xgbTree
48. Knn, J48, lda
49. Knn, J48, mlp
50. Knn, J48, earth
51. Knn, J48, rf
52. Knn, J48, rotationForest

53. Knn, J48, gbm
54. Knn, J48, svmLinear
55. Knn, J48, svmPoly
56. Knn, J48, svmRadial
57. Knn, J48, evtree
58. Knn, J48, glm
59. Knn, C5.0, xgbTree
60. Knn, C5.0, lda
61. Knn, C5.0, mlp
62. Knn, C5.0, earth
63. Knn, C5.0, rf
64. Knn, C5.0, rotationForest
65. Knn, C5.0, gbm
66. Knn, C5.0, svmLinear
67. Knn, C5.0, svmPoly
68. Knn, C5.0, svmRadial
69. Knn, C5.0, evtree
70. Knn, C5.0, glm
71. Knn, xgbTree, lda
72. Knn, xgbTree, mlp
73. Knn, xgbTree, earth
74. Knn, xgbTree, rf
75. Knn, xgbTree, rotationForest
76. Knn, xgbTree, gbm
77. Knn, xgbTree, svmLinear
78. Knn, xgbTree, svmPoly
79. Knn, xgbTree, svmRadial
80. Knn, xgbTree, evtree
81. Knn, xgbTree, glm
82. Knn, lda, mlp
83. Knn, lda, earth
84. Knn, lda, rf
85. Knn, lda, rotationForest
86. Knn, lda, gbm
87. Knn, lda, svmLinear
88. Knn, lda, svmPoly
89. Knn, lda, svmRadial
90. Knn, lda, evtree
91. Knn, lda, glm
92. Knn, mlp, earth
93. Knn, mlp, rf
94. Knn, mlp, rotationForest
95. Knn, mlp, gbm
96. Knn, mlp, svmLinear
97. Knn, mlp, svmPoly
98. Knn, mlp, svmRadial
99. Knn, mlp, evtree
100. Knn, mlp, glm
101. Knn, earth, rf
102. Knn, earth, rotationForest
103. Knn, earth, gbm
104. Knn, earth, svmLinear
105. Knn, earth, svmPoly
106. Knn, earth, svmRadial

107. Knn, earth, evtree
108. Knn, earth, glm
109. Knn, rf, rotationForest
110. Knn, rf, gbm
111. Knn, rf, svmLinear
112. Knn, rf, svmPoly
113. Knn, rf, svmRadial
114. Knn, rf, evtree
115. Knn, rf, glm
116. Knn, rotationForest, gbm
117. Knn, rotationForest, svmLinear
118. Knn, rotationForest, svmPoly
119. Knn, rotationForest, svmRadial
120. Knn, rotationForest, evtree
121. Knn, rotationForest, glm
122. Knn, gbm, svmLinear
123. Knn, gbm, svmPoly
124. Knn, gbm, svmRadial
125. Knn, gbm, evtree
126. Knn, gbm, glm
127. Knn, svmLinear, svmPoly
128. Knn, svmLinear, svmRadial
129. Knn, svmLinear, evtree
130. Knn, svmLinear, glm
131. Knn, svmPoly, svmRadial
132. Knn, svmPoly, evtree
133. Knn, svmPoly, glm
134. Knn, svmRadial, evtree
135. Knn, svmRadial, glm
136. Knn, evtree, glm
137. Nb, rpart, treebag
138. Nb, rpart, J48
139. Nb, rpart, C5.0
140. Nb, rpart, xgbTree
141. Nb, rpart, lda
142. Nb, rpart, mlp
143. Nb, rpart, earth
144. Nb, rpart, rf
145. Nb, rpart, rotationForest
146. Nb, rpart, gbm
147. Nb, rpart, svmLinear
148. Nb, rpart, svmPoly
149. Nb, rpart, svmRadial
150. Nb, rpart, evtree
151. Nb, rpart, glm
152. Nb, treebag, J48
153. Nb, treebag, C5.0
154. Nb, treebag, xgbTree
155. Nb, treebag, lda
156. Nb, treebag, mlp
157. Nb, treebag, earth
158. Nb, treebag, rf
159. Nb, treebag, rotationForest
160. Nb, treebag, gbm

161. Nb, treebag, svmLinear  
162. Nb, treebag, svmPoly  
163. Nb, treebag, svmRadial  
164. Nb, treebag, evtree  
165. Nb, treebag, glm  
166. Nb, J48, C5.0  
167. Nb, J48, xgbTree  
168. Nb, J48, lda  
169. Nb, J48, mlp  
170. Nb, J48, earth  
171. Nb, J48, rf  
172. Nb, J48, rotationForest  
173. Nb, J48, gbm  
174. Nb, J48, svmLinear  
175. Nb, J48, svmPoly  
176. Nb, J48, svmRadial  
177. Nb, J48, evtree  
178. Nb, J48, glm  
179. Nb, C5.0, xgbTree  
180. Nb, C5.0, lda  
181. Nb, C5.0, mlp  
182. Nb, C5.0, earth  
183. Nb, C5.0, rf  
184. Nb, C5.0, rotationForest  
185. Nb, C5.0, gbm  
186. Nb, C5.0, svmLinear  
187. Nb, C5.0, svmPoly  
188. Nb, C5.0, svmRadial  
189. Nb, C5.0, evtree  
190. Nb, C5.0, glm  
191. Nb, xgbTree, lda  
192. Nb, xgbTree, mlp  
193. Nb, xgbTree, earth  
194. Nb, xgbTree, rf  
195. Nb, xgbTree, rotationForest  
196. Nb, xgbTree, gbm  
197. Nb, xgbTree, svmLinear  
198. Nb, xgbTree, svmPoly  
199. Nb, xgbTree, svmRadial  
200. Nb, xgbTree, evtree  
201. Nb, xgbTree, glm  
202. Nb, lda, mlp  
203. Nb, lda, earth  
204. Nb, lda, rf  
205. Nb, lda, rotationForest  
206. Nb, lda, gbm  
207. Nb, lda, svmLinear  
208. Nb, lda, svmPoly  
209. Nb, lda, svmRadial  
210. Nb, lda, evtree  
211. Nb, lda, glm  
212. Nb, mlp, earth  
213. Nb, mlp, rf  
214. Nb, mlp, rotationForest

215. Nb, mlp, gbm  
216. Nb, mlp, svmLinear  
217. Nb, mlp, svmPoly  
218. Nb, mlp, svmRadial  
219. Nb, mlp, evtree  
220. Nb, mlp, glm  
221. Nb, earth, rf  
222. Nb, earth, rotationForest  
223. Nb, earth, gbm  
224. Nb, earth, svmLinear  
225. Nb, earth, svmPoly  
226. Nb, earth, svmRadial  
227. Nb, earth, evtree  
228. Nb, earth, glm  
229. Nb, rf, rotationForest  
230. Nb, rf, gbm  
231. Nb, rf, svmLinear  
232. Nb, rf, svmPoly  
233. Nb, rf, svmRadial  
234. Nb, rf, evtree  
235. Nb, rf, glm  
236. Nb, rotationForest, gbm  
237. Nb, rotationForest, svmLinear  
238. Nb, rotationForest, svmPoly  
239. Nb, rotationForest, svmRadial  
240. Nb, rotationForest, evtree  
241. Nb, rotationForest, glm  
242. Nb, gbm, svmLinear  
243. Nb, gbm, svmPoly  
244. Nb, gbm, svmRadial  
245. Nb, gbm, evtree  
246. Nb, gbm, glm  
247. Nb, svmLinear, svmPoly  
248. Nb, svmLinear, svmRadial  
249. Nb, svmLinear, evtree  
250. Nb, svmLinear, glm  
251. Nb, svmPoly, svmRadial  
252. Nb, svmPoly, evtree  
253. Nb, svmPoly, glm  
254. Nb, svmRadial, evtree  
255. Nb, svmRadial, glm  
256. Nb, evtree, glm  
257. Rpart, treebag, J48  
258. Rpart, treebag, C5.0  
259. Rpart, treebag, xgbTree  
260. Rpart, treebag, lda  
261. Rpart, treebag, mlp  
262. Rpart, treebag, earth  
263. Rpart, treebag, rf  
264. Rpart, treebag, rotationForest  
265. Rpart, treebag, gbm  
266. Rpart, treebag, svmLinear  
267. Rpart, treebag, svmPoly  
268. Rpart, treebag, svmRadial

269. Rpart, treebag, evtree  
270. Rpart, treebag, glm  
271. Rpart, J48, C5.0  
272. Rpart, J48, xgbTree  
273. Rpart, J48, lda  
274. Rpart, J48, mlp  
275. Rpart, J48, earth  
276. Rpart, J48, rf  
277. Rpart, J48, rotationForest  
278. Rpart, J48, gbm  
279. Rpart, J48, svmLinear  
280. Rpart, J48, svmPoly  
281. Rpart, 48, svmRadial  
282. Rpart, J48, evtree  
283. Rpart, J48, glm  
284. Rpart, C5.0, xgbTree  
285. Rpart, C5.0, lda  
286. Rpart, C5.0, mlp  
287. Rpart, C5.0, earth  
288. Rpart, C5.0, rf  
289. Rpart, C5.0, rotationForest  
290. Rpart, C5.0, gbm  
291. Rpart, C5.0, svmLinear  
292. Rpart, C5.0, svmPoly  
293. Rpart, C5.0, svmRadial  
294. Rpart, C5.0, evtree  
295. Rpart, C5.0, glm  
296. Rpart, xgbTree, lda  
297. Rpart, xgbTree, mlp  
298. Rpart, xgbTree, earth  
299. Rpart, xgbTree, rf  
300. Rpart, xgbTree, rotationForest  
301. Rpart, xgbTree, gbm  
302. Rpart, xgbTree, svmLinear  
303. Rpart, xgbTree, svmPoly  
304. Rpart, xgbTree, svmRadial  
305. Rpart, xgbTree, evtree  
306. Rpart, xgbTree, glm  
307. Rpart, lda, mlp  
308. Rpart, lda, earth  
309. Rpart, lda, rf  
310. Rpart, lda, rotationForest  
311. Rpart, lda, gbm  
312. Rpart, lda, svmLinear  
313. Rpart, lda, svmPoly  
314. Rpart, lda, svmRadial  
315. Rpart, lda, evtree  
316. Rpart, lda, glm  
317. Rpart, mlp, earth  
318. Rpart, mlp, rf  
319. Rpart, mlp, rotationForest  
320. Rpart, mlp, gbm  
321. Rpart, mlp, svmLinear  
322. Rpart, mlp, svmPoly

323. Rpart, mlp, svmRadial  
324. Rpart, mlp, evtree  
325. Rpart, mlp, glm  
326. Rpart, earth, rf  
327. Rpart, earth, rotationForest  
328. Rpart, earth, gbm  
329. Rpart, earth, svmLinear  
330. Rpart, earth, svmPoly  
331. Rpart, earth, svmRadial  
332. Rpart, earth, evtree  
333. Rpart, earth, glm  
334. Rpart, rf, rotationForest  
335. Rpart, rf, gbm  
336. Rpart, rf, svmLinear  
337. Rpart, rf, svmPoly  
338. Rpart, rf, svmRadial  
339. Rpart, rf, evtree  
340. Rpart, rf, glm  
341. Rpart, rotationForest, gbm  
342. Rpart, rotationForest, svmLinear  
343. Rpart, rotationForest, svmPoly  
344. Rpart, rotationForest, svmRadial  
345. Rpart, rotationForest, evtree  
346. Rpart, rotationForest, glm  
347. Rpart, gbm, svmLinear  
348. Rpart, gbm, svmPoly  
349. Rpart, gbm, svmRadial  
350. Rpart, gbm, evtree  
351. Rpart, gbm, glm  
352. Rpart, svmLinear, svmPoly  
353. Rpart, svmLinear, svmRadial  
354. Rpart, svmLinear, evtree  
355. Rpart, svmLinear, glm  
356. Rpart, svmPoly, svmRadial  
357. Rpart, svmPoly, evtree  
358. Rpart, svmPoly, glm  
359. Rpart, svmRadial, evtree  
360. Rpart, svmRadial, glm  
361. Rpart, evtree, glm  
362. Treebag, J48, C5.0  
363. Treebag, J48, xgbTree  
364. Treebag, J48, lda  
365. Treebag, J48, mlp  
366. Treebag, J48, earth  
367. Treebag, J48, rf  
368. Treebag, J48, rotationForest  
369. Treebag, J48, gbm  
370. Treebag, J48, svmLinear  
371. Treebag, J48, svmPoly  
372. Treebag, J48, svmRadial  
373. Treebag, J48, evtree  
374. Treebag, J48, glm  
375. Treebag, C5.0, xgbTree  
376. Treebag, C5.0, lda

377. Treebag, C5.0, mlp
378. Treebag, C5.0, earth
379. Treebag, C5.0, rf
380. Treebag, C5.0, rotationForest
381. Treebag, C5.0, gbm
382. Treebag, C5.0, svmLinear
383. Treebag, C5.0, svmPoly
384. Treebag, C5.0, svmRadial
385. Treebag, C5.0, evtree
386. Treebag, C5.0, glm
387. Treebag, xgbTree, lda
388. Treebag, xgbTree, mlp
389. Treebag, xgbTree, earth
390. Treebag, xgbTree, rf
391. Treebag, xgbTree, rotationForest
392. Treebag, xgbTree, gbm
393. Treebag, xgbTree, svmLinear
394. Treebag, xgbTree, svmPoly
395. Treebag, xgbTree, svmRadial
396. Treebag, xgbTree, evtree
397. Treebag, xgbTree, glm
398. Treebag, lda, mlp
399. Treebag, lda, earth
400. Treebag, lda, rf
401. Treebag, lda, rotationForest
402. Treebag, lda, gbm
403. Treebag, lda, svmLinear
404. Treebag, lda, svmPoly
405. Treebag, lda, svmRadial
406. Treebag, lda, evtree
407. Treebag, lda, glm
408. Treebag, mlp, earth
409. Treebag, mlp, rf
410. Treebag, mlp, rotationForest
411. Treebag, mlp, gbm
412. Treebag, mlp, svmLinear
413. Treebag, mlp, svmPoly
414. Treebag, mlp, svmRadial
415. Treebag, mlp, evtree
416. Treebag, mlp, glm
417. Treebag, earth, rf
418. Treebag, earth, rotationForest
419. Treebag, earth, gbm
420. Treebag, earth, svmLinear
421. Treebag, earth, svmPoly
422. Treebag, earth, svmRadial
423. Treebag, earth, evtree
424. Treebag, earth, glm
425. Treebag, rf, rotationForest
426. Treebag, rf, gbm
427. Treebag, rf, svmLinear
428. Treebag, rf, svmPoly
429. Treebag, rf, svmRadial
430. Treebag, rf, evtree

431. Treebag, rf, glm  
432. Treebag, rotationForest, gbm  
433. Treebag, rotationForest, svmLinear  
434. Treebag, rotationForest, svmPoly  
435. Treebag, rotationForest, svmRadial  
436. Treebag, rotationForest, evtree  
437. Treebag, rotationForest, glm  
438. Treebag, gbm, svmLinear  
439. Treebag, gbm, svmPoly  
440. Treebag, gbm, svmRadial  
441. Treebag, gbm, evtree  
442. Treebag, gbm, glm  
443. Treebag, svmLinear, svmPoly  
444. Treebag, svmLinear, svmRadial  
445. Treebag, svmLinear, evtree  
446. Treebag, svmLinear, glm  
447. Treebag, svmPoly, svmRadial  
448. Treebag, svmPoly, evtree  
449. Treebag, svmPoly, glm  
450. Treebag, svmRadial, evtree  
451. Treebag, svmRadial, glm  
452. Treebag, evtree, glm  
453. J48, C5.0, xgbTree  
454. J48, C5.0, lda  
455. J48, C5.0, mlp  
456. J48, C5.0, earth  
457. J48, C5.0, rf  
458. J48, C5.0, rotationForest  
459. J48, C5.0, gbm  
460. J48, C5.0, svmLinear  
461. J48, C5.0, svmPoly  
462. J48, C5.0, svmRadial  
463. J48, C5.0, evtree  
464. J48, C5.0, glm  
465. J48, xgbTree, lda  
466. J48, xgbTree, mlp  
467. J48, xgbTree, earth  
468. J48, xgbTree, rf  
469. J48, xgbTree, rotationForest  
470. J48, xgbTree, gbm  
471. J48, xgbTree, svmLinear  
472. J48, xgbTree, svmPoly  
473. J48, xgbTree, svmRadial  
474. J48, xgbTree, evtree  
475. J48, xgbTree, glm  
476. J48, lda, mlp  
477. J48, lda, earth  
478. J48, lda, rf  
479. J48, lda, rotationForest  
480. J48, lda, gbm  
481. J48, lda, svmLinear  
482. J48, lda, svmPoly  
483. J48, lda, svmRadial  
484. J48, lda, evtree

485. J48, lda, glm  
486. J48, mlp, earth  
487. J48, mlp, rf  
488. J48, mlp, rotationForest  
489. J48, mlp, gbm  
490. J48, mlp, svmLinear  
491. J48, mlp, svmPoly  
492. J48, mlp, svmRadial  
493. J48, mlp, evtree  
494. J48, mlp, glm  
495. J48, earth, rf  
496. J48, earth, rotationForest  
497. J48, earth, gbm  
498. J48, earth, svmLinear  
499. J48, earth, svmPoly  
500. J48, earth, svmRadial  
501. J48, earth, evtree  
502. J48, earth, glm  
503. J48, rf, rotationForest  
504. J48, rf, gbm  
505. J48, rf, svmLinear  
506. J48, rf, svmPoly  
507. J48, rf, svmRadial  
508. J48, rf, evtree  
509. J48, rf, glm  
510. J48, rotationForest, gbm  
511. J48, rotationForest, svmLinear  
512. J48, rotationForest, svmPoly  
513. J48, rotationForest, svmRadial  
514. J48, rotationForest, evtree  
515. J48, rotationForest, glm  
516. J48, gbm, svmLinear  
517. J48, gbm, svmPoly  
518. J48, gbm, svmRadial  
519. J48, gbm, evtree  
520. J48, gbm, glm  
521. J48, svmLinear, svmPoly  
522. J48, svmLinear, svmRadial  
523. J48, svmLinear, evtree  
524. J48, svmLinear, glm  
525. J48, svmPoly, svmRadial  
526. J48, svmPoly, evtree  
527. J48, svmPoly, glm  
528. J48, svmRadial, evtree  
529. J48, svmRadial, glm  
530. J48, evtree, glm  
531. C5.0, xgbTree, lda  
532. C5.0, xgbTree, mlp  
533. C5.0, xgbTree, earth  
534. C5.0, xgbTree, rf  
535. C5.0, xgbTree, rotationForest  
536. C5.0, xgbTree, gbm  
537. C5.0, xgbTree, svmLinear  
538. C5.0, xgbTree, svmPoly

539. C5.0, xgbTree, svmRadial  
540. C5.0, xgbTree, evtree  
541. C5.0, xgbTree, glm  
542. C5.0, lda, mlp  
543. C5.0, lda, earth  
544. C5.0, lda, rf  
545. C5.0, lda, rotationForest  
546. C5.0, lda, gbm  
547. C5.0, lda, svmLinear  
548. C5.0, lda, svmPoly  
549. C5.0, lda, svmRadial  
550. C5.0, lda, evtree  
551. C5.0, lda, glm  
552. C5.0, mlp, earth  
553. C5.0, mlp, rf  
554. C5.0, mlp, rotationForest  
555. C5.0, mlp, gbm  
556. C5.0, mlp, svmLinear  
557. C5.0, mlp, svmPoly  
558. C5.0, mlp, svmRadial  
559. C5.0, mlp, evtree  
560. C5.0, mlp, glm  
561. C5.0, earth, rf  
562. C5.0, earth, rotationForest  
563. C5.0, earth, gbm  
564. C5.0, earth, svmLinear  
565. C5.0, earth, svmPoly  
566. C5.0, earth, svmRadial  
567. C5.0, earth, evtree  
568. C5.0, earth, glm  
569. C5.0, rf, rotationForest  
570. C5.0, rf, gbm  
571. C5.0, rf, svmLinear  
572. C5.0, rf, svmPoly  
573. C5.0, rf, svmRadial  
574. C5.0, rf, evtree  
575. C5.0, rf, glm  
576. C5.0, rotationForest, gbm  
577. C5.0, rotationForest, svmLinear  
578. C5.0, rotationForest, svmPoly  
579. C5.0, rotationForest, svmRadial  
580. C5.0, rotationForest, evtree  
581. C5.0, rotationForest, glm  
582. C5.0, gbm, svmLinear  
583. C5.0, gbm, svmPoly  
584. C5.0, gbm, svmRadial  
585. C5.0, gbm, evtree  
586. C5.0, gbm, glm  
587. C5.0, svmLinear, svmPoly  
588. C5.0, svmLinear, svmRadial  
589. C5.0, svmLinear, evtree  
590. C5.0, svmLinear, glm  
591. C5.0, svmPoly, svmRadial  
592. C5.0, svmPoly, evtree

593. C5.0, svmPoly, glm  
594. C5.0, svmRadial, evtree  
595. C5.0, svmRadial, glm  
596. C5.0, evtree, glm  
597. XGBTREE, lda, mlp  
598. XGBTREE, lda, earth  
599. XGBTREE, lda, rf  
600. XGBTREE, lda, rotationForest  
601. XGBTREE, lda, gbm  
602. XGBTREE, lda, svmLinear  
603. XGBTREE, lda, svmPoly  
604. XGBTREE, lda, svmRadial  
605. XGBTREE, lda, evtree  
606. XGBTREE, lda, glm  
607. XGBTREE, mlp, earth  
608. XGBTREE, mlp, rf  
609. XGBTREE, mlp, rotationForest  
610. XGBTREE, mlp, gbm  
611. XGBTREE, mlp, svmLinear  
612. XGBTREE, mlp, svmPoly  
613. XGBTREE, mlp, svmRadial  
614. XGBTREE, mlp, evtree  
615. XGBTREE, mlp, glm  
616. XGBTREE, earth, rf  
617. XGBTREE, earth, rotationForest  
618. XGBTREE, earth, gbm  
619. XGBTREE, earth, svmLinear  
620. XGBTREE, earth, svmPoly  
621. XGBTREE, earth, svmRadial  
622. XGBTREE, earth, evtree  
623. XGBTREE, earth, glm  
624. XGBTREE, rf, rotationForest  
625. XGBTREE, rf, gbm  
626. XGBTREE, rf, svmLinear  
627. XGBTREE, rf, svmPoly  
628. XGBTREE, rf, svmRadial  
629. XGBTREE, rf, evtree  
630. XGBTREE, rf, glm  
631. XGBTREE, rotationForest, gbm  
632. XGBTREE, rotationForest, svmLinear  
633. XGBTREE, rotationForest, svmPoly  
634. XGBTREE, rotationForest, svmRadial  
635. XGBTREE, rotationForest, evtree  
636. XGBTREE, rotationForest, glm  
637. XGBTREE, gbm, svmLinear  
638. XGBTREE, gbm, svmPoly  
639. XGBTREE, gbm, svmRadial  
640. XGBTREE, gbm, evtree  
641. XGBTREE, gbm, glm  
642. XGBTREE, svmLinear, svmPoly  
643. XGBTREE, svmLinear, svmRadial  
644. XGBTREE, svmLinear, evtree  
645. XGBTREE, svmLinear, glm  
646. XGBTREE, svmPoly, svmRadial

647. XGBTREE, svmPoly, evtree  
648. XGBTREE, svmPoly, glm  
649. XGBTREE, svmRadial, evtree  
650. XGBTREE, svmRadial, glm  
651. XGBTREE, evtree, glm  
652. LDA, mlp, earth  
653. LDA, mlp, rf  
654. LDA, mlp, rotationForest  
655. LDA, mlp, gbm  
656. LDA, mlp, svmLinear  
657. LDA, mlp, svmPoly  
658. LDA, mlp, svmRadial  
659. LDA, mlp, evtree  
660. LDA, mlp, glm  
661. LDA, earth, rf  
662. LDA, earth, rotationForest  
663. LDA, earth, gbm  
664. LDA, earth, svmLinear  
665. LDA, earth, svmPoly  
666. LDA, earth, svmRadial  
667. LDA, earth, evtree  
668. LDA, earth, glm  
669. LDA, rf, rotationForest  
670. LDA, rf, gbm  
671. LDA, rf, svmLinear  
672. LDA, rf, svmPoly  
673. LDA, rf, svmRadial  
674. LDA, rf, evtree  
675. LDA, rf, glm  
676. LDA, rotationForest, gbm  
677. LDA, rotationForest, svmLinear  
678. LDA, rotationForest, svmPoly  
679. LDA, rotationForest, svmRadial  
680. LDA, rotationForest, evtree  
681. LDA, rotationForest, glm  
682. LDA, gbm, svmLinear  
683. LDA, gbm, svmPoly  
684. LDA, gbm, svmRadial  
685. LDA, gbm, evtree  
686. LDA, gbm, glm  
687. LDA, svmLinear, svmPoly  
688. LDA, svmLinear, svmRadial  
689. LDA, svmLinear, evtree  
690. LDA, svmLinear, glm  
691. LDA, svmPoly, svmRadial  
692. LDA, svmPoly, evtree  
693. LDA, svmPoly, glm  
694. LDA, svmRadial, evtree  
695. LDA, svmRadial, glm  
696. LDA, evtree, glm  
697. MLP, earth, rf  
698. MLP, earth, rotationForest  
699. MLP, earth, gbm  
700. MLP, earth, svmLinear

701. MLP, earth, svmPoly  
702. MLP, earth, svmRadial  
703. MLP, earth, evtree  
704. MLP, earth, glm  
705. MLP, rf, rotationForest  
706. MLP, rf, gbm  
707. MLP, rf, svmLinear  
708. MLP, rf, svmPoly  
709. MLP, rf, svmRadial  
710. MLP, rf, evtree  
711. MLP, rf, glm  
712. MLP, rotationForest, gbm  
713. MLP, rotationForest, svmLinear  
714. MLP, rotationForest, svmPoly  
715. MLP, rotationForest, svmRadial  
716. MLP, rotationForest, evtree  
717. MLP, rotationForest, glm  
718. MLP, gbm, svmLinear  
719. MLP, gbm, svmPoly  
720. MLP, gbm, svmRadial  
721. MLP, gbm, evtree  
722. MLP, gbm, glm  
723. MLP, svmLinear, svmPoly  
724. MLP, svmLinear, svmRadial  
725. MLP, svmLinear, evtree  
726. MLP, svmLinear, glm  
727. MLP, svmPoly, svmRadial  
728. MLP, svmPoly, evtree  
729. MLP, svmPoly, glm  
730. MLP, svmRadial, evtree  
731. MLP, svmRadial, glm  
732. MLP, evtree, glm  
733. EARTH, rf, rotationForest  
734. EARTH, rf, gbm  
735. EARTH, rf, svmLinear  
736. EARTH, rf, svmPoly  
737. EARTH, rf, svmRadial  
738. EARTH, rf, evtree  
739. EARTH, rf, glm  
740. EARTH, rotationForest, gbm  
741. EARTH, rotationForest, svmLinear  
742. EARTH, rotationForest, svmPoly  
743. EARTH, rotationForest, svmRadial  
744. EARTH, rotationForest, evtree  
745. EARTH, rotationForest, glm  
746. EARTH, gbm, svmLinear  
747. EARTH, gbm, svmPoly  
748. EARTH, gbm, svmRadial  
749. EARTH, gbm, evtree  
750. EARTH, gbm, glm  
751. EARTH, svmLinear, svmPoly  
752. EARTH, svmLinear, svmRadial  
753. EARTH, svmLinear, evtree  
754. EARTH, svmLinear, glm

755. EARTH, svmPoly, svmRadial  
756. EARTH, svmPoly, evtree  
757. EARTH, svmPoly, glm  
758. EARTH, svmRadial, evtree  
759. EARTH, svmRadial, glm  
760. EARTH, evtree, glm  
761. RF, rotationForest, gbm  
762. RF, rotationForest, svmLinear  
763. RF, rotationForest, svmPoly  
764. RF, rotationForest, svmRadial  
765. RF, rotationForest, evtree  
766. RF, rotationForest, glm  
767. RF, gbm, svmLinear  
768. RF, gbm, svmPoly  
769. RF, gbm, svmRadial  
770. RF, gbm, evtree  
771. RF, gbm, glm  
772. RF, svmLinear, svmPoly  
773. RF, svmLinear, svmRadial  
774. RF, svmLinear, evtree  
775. RF, svmLinear, glm  
776. RF, svmPoly, svmRadial  
777. RF, svmPoly, evtree  
778. RF, svmPoly, glm  
779. RF, svmRadial, evtree  
780. RF, svmRadial, glm  
781. RF, evtree, glm  
782. ROTATIONFOREST, gbm, svmLinear  
783. ROTATIONFOREST, gbm, svmPoly  
784. ROTATIONFOREST, gbm, svmRadial  
785. ROTATIONFOREST, gbm, evtree  
786. ROTATIONFOREST, gbm, glm  
787. ROTATIONFOREST, svmLinear, svmPoly  
788. ROTATIONFOREST, svmLinear, svmRadial  
789. ROTATIONFOREST, svmLinear, evtree  
790. ROTATIONFOREST, svmLinear, glm  
791. ROTATIONFOREST, svmPoly, svmRadial  
792. ROTATIONFOREST, svmPoly, evtree  
793. ROTATIONFOREST, svmPoly, glm  
794. ROTATIONFOREST, svmRadial, evtree  
795. ROTATIONFOREST, svmRadial, glm  
796. ROTATIONFOREST, evtree, glm  
797. GBM, svmLinear, svmPoly  
798. GBM, svmLinear, svmRadial  
799. GBM, svmLinear, evtree  
800. GBM, svmLinear, glm  
801. GBM, svmPoly, svmRadial  
802. GBM, svmPoly, evtree  
803. GBM, svmPoly, glm  
804. GBM, svmRadial, evtree  
805. GBM, svmRadial, glm  
806. GBM, evtree, glm  
807. SVMLINEAR, svmPoly, svmRadial  
808. SVMLINEAR, svmPoly, evtree

- 809. SVMLINEAR, svmPoly, glm
- 810. SVMLINEAR, svmRadial, evtree
- 811. SVMLINEAR, svmRadial, glm
- 812. SVMLINEAR, evtree, glm
- 813. SVMPOLY, svmRadial, evtree
- 814. SVMPOLY, svmRadial, glm
- 815. SVMPOLY, evtree, glm
- 816. SVMRADIAL, evtree, glm

