T.C.

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

GERÇEK VERİ SETLERİNDE SINIFLANDIRMA YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

RAMAZAN AYÖZ

OCAK 2021

MUĞLA

R. AYÖZ İSTATİSTİK ANABİLİM DALI YÜKSEK LİSANS TEZİ MUĞLA 2021

T.C.

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK ANABİLİM DALI

GERÇEK VERİ SETLERİNDE SINIFLANDIRMA YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

RAMAZAN AYÖZ

OCAK 2021

MUĞLA

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

RAMAZAN AYÖZ tarafından hazırlanan **GERÇEK VERİ SETLERİNDE SINIFLANDIRMA YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI** başlıklı tezinin 05/01/2021 tarihinde aşağıdaki jüri tarafından İstatistik Anabilim Dalı'nda yüksek lisans derecesi için geçerli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JÜRİSİ

Prof. Dr. Öznur İŞÇİ GÜNERİ (Jüri Başkanı) İmza:
İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Nevin GÜLER DİNCER (Danışman) İmza:
İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Dr. Öğr. Üyesi Aynur İNCEKIRIK (Üye) İmza:
İstatistik Anabilim Dalı,
Celal Bayar Üniversitesi, Manisa

ANA BİLİM DALI BAŞKANLIĞI ONAYI

Prof. Dr. Dursun AYDIN İmza
İstatistik Ana Bilim Dalı Başkanı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Nevin GÜLER DİNCER İmza
Danışman, İstatistik Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Tez çalışmam sırasında tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş, başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Ramazan AYÖZ

05/01/2021

ÖZET

GERÇEK VERİ SETLERİNDE SINIFLANDIRMA YÖNTEMLERİNİN PERFORMANSLARININ KARŞILAŞTIRILMASI

Ramazan AYÖZ

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalı

Danışman: Doç. Dr. Nevin GÜLER DİNCER

Ocak 2021, 85 sayfa

Veri madenciliğinde sınıflandırma, çıktı(sınıf) değerleri bilinen gözlemler kullanılarak makine öğrenmesi yöntemleri ile bir model oluşturulması ve bu modelin daha sonra sınıf değeri bilinmeyen gözlemlerin sınıf değerlerini tahmin etmek amacıyla kullanılması olarak tanımlanabilir. Bu süreçte uygulanabilecek farklı sınıflandırma algoritmaları bulunur ve bu algoritmaların başarılarını farklı kriterler üzerinden incelemek mümkündür. Sınıflandırma, tahmine dayalı bir yöntem olduğu için en güçlü kriterin algoritmanın bir sınıfı doğru tahmin etme olasılığı olduğunu söylemek mümkündür. Bu yüzden tüm test gözlemleri içerisinde doğru sınıflandırılmış gözlem oranlarını incelemek sınıflandırma başarısını gösteren en önemli kriterlerden biridir.

Bu çalışmanın ana konusu, doğruluk kriteri kullanılarak WEKA veri madenciliği yazılımında bulunan 41 farklı sınıflandırma yönteminin gerçek ve simulasyon çalışması ile üretilen veri setlerini sınıflandırmadaki performanslarını karşılaştırmaktadır. Bu amaca yönelik olarak farklı alanlara ait değişik gözlem, değişken ve sınıf sayılarına sahip 100 gerçek veri seti, yine farklı yapılarda üretilen 100 simulasyon veri seti kullanılmıştır.

Bu çalışma sonucunda, hiçbir sınıflandırıcının her veri seti için en iyi performansı sergilemediği ve tüm veri setlerinde en iyi performansı yakalamak amacıyla farklı sınıflandırıcıların test edilmesi gerektiği görülmüştür. Ancak tüm sınıflandırıcılar içerisinde doğru sınıflandırılmış örnek oranları içerisinde değişimi en az olan ve en optimal şekilde en iyi sonuçları veren algoritmaların karar ağaçları tabanlı bir sınıflandırma algoritması olan Random Forest, karar ağaçları ve lojistiğin birleştirildiği bir algoritma olan LMT ve lojistik tabanlı bir sınıflandırıcı olan Logit Boost algoritması olduğu görülmüştür. Aynı zamanda gerçek veri setlerinin başarılarının simulasyon veri setlerinin başarılarından daha üstün olduğu görülmüştür.

Anahtar Kelimeler: Veri Madenciliği, Sınıflandırma Algoritmaları, Karar Ağaçları, Lojistic Regresyon

ABSTRACT

COMPARISION OF PERFORMANCE OF CLASSIFICATION METHODS IN REAL DATA SETS

Ramazan AYOZ

Master of Science(M. Sc.)

Graduate School of Natural and Applied Science

Department of Statistics

Supervisior: Assoc. Prof. Dr. Nevin GULER DINCER

January 2021, 85 page

Classification in data mining can be defined as creating a model with machine learning methods using observations with known output(class) values, and then using this model to predict the class values of observations with unknown class values. There are different classification algorithms that can be applied in this process and it is possible to examine the success of these algorithms on different criteria. Since the classification is a method based on prediction, it is possible to say that the most powerful criterion is the probability of the algorithm to correctly predict a class. Therefore, examining the proportion of correctly classified observations among all test observations is one of the most important criteria showing the success of classification.

The main subject of this thesis is to compare the performances of 41 different classification methods existed in WEKA data mining software in classifying the real data sets and the data sets generated via simulation studies by using accuracy criterion. For this purpose, 100 real data sets with different number of observations, variables and class and 100 simulation data sets generated in different structures are used.

As a result of this study, it is seen that no classifier has the best performance for each data set and different classifiers should be tested in order to achieve the best performance in all data sets. However, it is seen that Random Forest, which is a decision tree-based classification algorithm, LMT, which is an algorithm combining decision trees and logistics, and Logit Boost algorithm, which is a logistics-based classifier have the least deviation among correctly classified sample rates among all classifiers. At the same time, it was observed that the success of real data sets is superior to the success of simulation data sets.

Keywords: Data Mining, Classification Algorithms, Decision Trees, Logistic Regression

ÖNSÖZ

Öncelikle lisans eğitimim süresince bana olan güveniyle kendime inanmamı sağlayan, yüksek lisans sürecimin tamamında bilgisini ve ilgisini hiç esirgemeyen, bu tez çalışmasının her aşamasında desteğini ve tecrübesini paylaşan, yol gösteren, bugün olduğum yerde olmamı sağlayan tez danışmanım Doç. Dr. Nevin GÜLER DİNCER'e;

Yine lisans ve yüksek lisans eğitimimde bilgileriniyle beni donatan değerli bölüm hocalarıma sonsuz teşekkürlerimi bir borç bilirim.

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

SEMBOLLER VE KISALTMALAR DİZİNİ

μ_X	Özniteliğe ait ortalama değer
σ_X	Özniteliğe ait standart sapma değeri
x_i^1	Normalize edilmiş gözlem
x_i	Gözlem değeri, bağımsız değişkenlerden birisi
x_{min}	Öznitelik içerisindeki en küçük değer
x_{max}	Öznitelik içerisindeki en büyük değer
p_i	Öznitelik değerleri içerisinde i gözleminin görülme sıklığı
S_v	V değerine sahip gözlemler
E(S)	S özniteliğine ait entropi değeri
V(A)	A değişkenini içeren değerler kümesi
$Gini_{Sağ}$	Sağ gruba ait Gini indeksi değeri
$Gini_{Sol}$	Sol gruba ait Gini indeksi değeri
n	Sınıf sayısı
$D_{Sağ}$	Gini indeksi hesabında özelliğe ait hesaplanan ortalamadan büyük veya eşit olan gözlem sayısı
D_{Sol}	Gini indeksi hesabında özelliğe ait hesaplanan ortalamadan küçük olan gözlem sayısı
R_i	Gini indeksi hesabında sağ grupta bulunan ve i sınıfına ait olan gözlem sayısı
L_i	Gini indeksi hesabında sol grupta bulunan ve i sınıfına ait gözlem sayısı
k	Eğitim veri setine ait gözlem sayısı
$G_{Sağ}$	Sağ gruba ait Gini İndeksi değeri
G_{Sol}	Sol gruba ait Gini İndeksi değeri
P(B\A)	A olayı gerçekleştiğinde B olayının da gerçekleşmesi olasılığı
P(A\B)	B olayı gerçekleştiğinde A olayının da gerçekleşmesi olasılığı
P(B)	B olayının marjinal olasılık değeri
P(A)	A olayının marjinal olasılık değeri
C_i	Sınıf özniteliği içerisindeki i sınıfı
X_k	Özellikler vektörü içerisinde k özelliği
w_{ij}	Bir bağlantının oluşturduğu ağırlık değeri

DVM	Destek Vektör Makineleri
DSÖO	Doğru Sınıflandırılmış Örnek Oranı
MSE	Ortalama Mutlak Hata
SVM	Support Vector Machines
NFL	No Free Lunch
DP	Doğru bir şekilde pozitif olarak sınıflandırılmış gözlemler
DN	Doğru bir şekilde negatif olarak sınıflandırılmış gözlemler
YP	Yanlış bir şekilde pozitif olarak sınıflandırılmış gözlemler
YN	Yanlış bir şekilde negatif olarak sınıflandırılmış gözlemler
NET	Yapay sinir ağları içerisindeki gizli nöronun sahip olduğu fonksiyon

1.GİRİŞ

Geçmişten günümüze sürekli olarak gelişen teknoloji ve internet erişimi sayesinde, farklı alanlarda veri elde etmek, bunları depolamak ve bu verilere erişmek mümkün hâle gelmektedir. Bu verilerin bir kısmı üzerinde hiçbir işlem yapılmadan, bir kısmı ise belli bir amaç doğrultusunda işlendikten sonra depolanır. Depolanan veriler daha sonra farklı yöntemler kullanılarak, yeni bilgilerin ortaya çıkarılması, tahmin edilmesi gibi amaçlarla kullanılabilir. Örnek olarak, bankalar müşteri profili oluşturup kredi başvurusunda bulunan kişiler için yeterliliği ölçebilir, marketler mağazadaki ürünlerin raf dizilimini belirlemek için birlikte satın alınan ürünleri inceleyebilir, belirli semptomlara sahip kişilerin hastalığının ne olduğu tespit edilebilir.

Veri setlerindeki gözlem sayısı küçük olduğunda gözlem yaparak örnek verilen çıkarımları yapmak mümkündür ancak çok büyük hacimli veri setlerinde bu pek mümkün olmamaktadır. Bu şekilde büyük miktardaki verilerden oluşan veri setlerinden, gözlemleme yoluyla elde edemeyeceğimiz bilgilerin ortaya çıkarılması veri madenciliği sayesinde mümkün olmaktadır. Veri madenciliği, çok miktarda veriye sahip olan veri setlerinden farklı kalıpları ve gizli bilgileri keşfetme sürecidir (Han v.d., 2011). Bu süreçte hedeflenen amaca uygun olarak seçilebilmesi gereken farklı veri madenciliği yöntemleri vardır. Bu yöntemleri sınıflandırma, kümeleme analizi ve birliktelik analizi şeklinde 3 ana başlık altında toplamak mümkündür. Her üç yöntem kısaca şu şekilde özetlenebilir. Kümeleme analizi k sayıda değişken ve n sayıda bireyden oluşan bir veri seti verildiğinde, değişkenler bakımından benzer ve farklı davranan bireylerin belirlenmesi şeklinde özetlenebilir. Adından da anlaşılacağı üzere, kümeleme analizi birbirine benzer bireylerden oluşan kümelerin oluşturulmasını hedefler. Birliktelik analizinde ise amaç, veri setinde birlikte hareket eden, aralarında bağlantı, bağıntı bulunan özniteliklerin (değişkenlerin) belirlenmesidir.

Bu tezin ana konusunu sınıflandırma oluşturmaktadır. Sınıflandırma, içerisinde sınıf etiketi bulunan veri setlerine yeni eklenecek ve sınıf değeri bilinmeyen gözlemlerin sınıf etiketlerinin tahmin edilmesi sürecidir. Şu ana kadar ana başlıkları, karar ağacı, Bayes, destek vektör makineleri, yapay sinir ağları şeklinde verilebilecek çok sayıda sınıflandırma yöntemi geliştirilmiştir. Sınıflandırmadan elde edilecek başarının arttırılabilmesi için öncelikle veri setine uygun sınıflandırma yönteminin seçilmesi gerekmektedir. Sınıflandırma algoritmalarının başarılarının ölçülmesi konusunda algoritmanın işlemi tamamlama süresi gibi kriterler incelenebilmesine karşın en önemli etken doğru sınıflandırılmış örnek oranlarıdır. Çünkü veri madenciliğinde sınıflandırmanın amacı sınıf değişkeni bilinmeyen bir gözlemin sınıfını tahmin etmektir ve bu tahminin olabildiğince başarılı olması istenir. Bu amaçla literatürde yer alan, sınıflandırma algoritmalarının başarılarını doğru sınıflandırılmış örnek oranları üzerinden karşılaştıran bazı çalışmalar aşağıda verilmiştir.

Mhetre ve Nagar (2017), Hindistan'da bulunan Mumbai Üniversitesi'ne bağlı Sardar Patel Teknoloji Enstitüsü öğrencilerine ait bir veri setini kullanarak öğrencilerin performanslarını tahmin edilmesinde yararlı olabilecek kalıpları belirlemek ve özellikle daha az başarılı olan öğrencilerin belirlenerek o öğrencilere özel destek kararı verilmesini sağlamak amacıyla çalışma yürütmüştür. Bu çalışmada Weka veri madenciliği yazılımda bulunan ZeroR, Naive Bayes, J48 ve Random Tree sınıflandırma algoritmaları ile sınıflandırıcıların başarılarını test etmiş, sonuç olarak da %95.4545 doğru sınıflandırma oranıyla en yüksek orana sahip olan algoritmayı ağaç tabanlı olan Random Tree olarak bulmuştur. En düşük doğru sınıflandırma oranına sahip algoritma ise %36.3636 ile ZeroR olmuştur.

Hassan vd. (2018), kalp ve hepatit rahatsızlıklarına ait iki adet veri seti üzerinde çalışarak bu hastalıkların yüksekliğini en iyi şekilde tahmin edebilecek sınıflandırma algoritmasını bulmayı amaçlamışlardır. Bu çalışmada Lojistik, Karar Ağacı, Naive Bayes, KNN, Destek Vektör Makineleri (DVM) ve Rasgele Orman algoritmalarını kullanmışlardır. Sınıflandırıcıların başarıları doğruluk, kesinlik, duyarlılık ve F değerleri bakımından karşılaştırılmış ve iki veri seti için de Rasgele Orman algoritmasının, başarı kriterlerinin hepsi için en iyi sonucu verdiği görülmüştür.

Verma ve Mishra (2017), UCI veri havuzundan elde ettikleri göğüs kanseri ve diyabet bilgilerini içeren iki adet veri seti ile Navie Bayes, J48, Çok Karmanlı PErceptron, SMO ve Reptree algoritmalarının başarılarını doğru sınıflandırılmış örnek oranı, yanlış sınıflandırılmış örnek oranı ve modelin oluşturulması için geçen süre bakımından incelemişlerdir. İki veri seti için de model oluşturma süresi bakımından en verimli sonuçları veren Naive Bayes sınıflandırıcısı olarak bulunmuştur. Göğüs kanserine ait veri setinde en yüksek başarıyı J48 sınıflandırıcısı sağlarken, diyabete ait veri setinde ise en yüksek başarıyı SMO sınıflandırıcısının sağladığını gözlemlemişlerdir.

Osisanwo vd. (2017), UCI veri havuzundan elde ettikleri 9 özniteliğe ve 768 gözleme sahip veri seti ve yine bu veri seti içerisinden elde edilen 6 özniteliğe ve 384 gözleme sahip veri seti ile çalışmışlardır. Bu süreçte 7 adet sınıflandırıcının başarıları doğru sınıflandırılmış örnek oranı, modelin oluşturulması için geçen süre, Kappa istatistiği ve ortalama mutlak hata(MSE) değerleri açısından karşılaştırılmıştır. Sınıflandırma uygulanırken 10 kat çapraz doğrulama yöntemi kullanılmıştır. Karşılaştırma sonucunda 768 gözleme sahip büyük veri seti için DVM sınıflandırıcısının en yüksek doğru sınıflandırma oranına, en yüksek Kappa değerine ve en düşük ortalama mutlak hata değerine sahip olarak en başarılı algoritma olduğu gözlemlenmiştir. 384 gözleme sahip alt veri seti için ise en yüksek doğru sınıflandırma oranına ve en düşük ortalama mutlak hata değerine sahip algoritma olarak DVM, en yüksek Kappa istatistiği değerine sahip algoritma olarak Rasgele Orman gözlemlenmiştir. Çalışma sonucunda Karar Ağacı ve Karar Tablosu sınıflandırıcılarının en düşük doğru sınıflandırılmış örnek oranı değerine sahip oldukları, DVM ve Naive Bayes sınıflandırıcılarının ise gözlem sayısı arttıkça daha iyi sonuçlar verdiği gözlemlenmiştir. Osisanwo vd. (2017) bu çalışma sonucunda, sınıflandırma algoritmalarından alınacak verimin arttırılabilmesi için büyük veri setleriyle çalışılmasını önermişlerdir.

Ali ve Smith (2006), UCI veri havuzundan ve Knowledge Discovery Central'dan elde ettikleri 100 adet veri seti ile 8 adet sınıflandırma algoritmasının başarılarını çeşitli doğruluk ve karmaşıklık kriterleri ile test etmişlerdir. Bu çalışma yapılırken No Free Lunch(NFL) teoreminden de bahseder. Bu teorem "hiçbir yöntem her durumda diğerlerinden daha iyi performans göstermez" düşüncesini savunmaktadır.

Çalışma sonucunda No Free Lunch teoremini destekler şekilde veri setleri üzerinde hiçbir algoritmanın tek tip olarak en başarılı algoritma olmadığı görülmüştür. Ali ve Smith (2006) sınıflandırma sürecinde 1000'den az gözleme sahip veri setlerinde 10 kat çapraz doğrulama yöntemini, 1000'den fazla gözleme sahip veri setlerinde ise veri setini %70 eğitim ve %30 test verisi olarak ayırarak çalışmışlardır. Sınıflandırıcıların başarıları karşılaştırılırken doğru sınıflandırılmış örnek oranlarının yanı sıra hem eğitim hem de test sürelerini karşılaştırdılar ve OneR sınıflandırıcısının model oluşturma süresinde en başarılı sınıflandırıcı olmasına rağmen doğru sınıflandırılmış örnek oranı bakımından en başarısız sınıflandırıcı olduğunu gördüler. Araştırma sonucunda C4.5, Yapay Sinir Ağları ve DVM sınıflandırıcılarının en iyi sonuçları verdiği gözlemlenmiştir.

Nookala vd. (2013), sınıflandırma algoritmalarının başarılarını karşılaştırmak için kanser hastalığı verilerini içeren 3 adet veri seti ve 14 adet sınıflandırıcı ile çalışmışlardır. Çalışma içerisinde hiçbir sınıflandırıcının diğer tüm sınıflandırıcılardan daha iyi performans göstermediği görülmüştür. Aynı zamanda gözlem sayısı arttıkça sınıflandırma algoritmalarının çoğunun daha fazla başarı gösterme eğiliminde olduğu görülmüştür ve kullanıcılara bir sınıflandırıcıya bağlı kalmayıp, farklı algoritmaları değerlendirmeleri gerektiğini belirtmiştir. Çok katmanlı perceptron sınıflandırıcısının tüm veri setleri içerisinde sadece birinde yüksek başarı gösterdiği ve model oluşturma süresinin diğer algoritmalara kıyasla çok daha uzun sürdüğü görülmüştür.

Zhongguo vd. (2017), geçmiş çalışmalardan yola çıkarak tek bir sınıflandırma algoritmasının her veri setinde en iyi performansı göstermeyeceği kanaatindedir ve No Free Lunch teoreminin tutarlı olduğunu savunmaktadır. 100 gerçek veri seti ve 9 sınıflandırıcı ile çalışan Zhongguo vd. (2017) öncelikle 10 kat çapraz doğrulama yöntemini varsayılan parametreler ile test etmiş, ardından farklı parametreler kullanarak karşılaştırma yapmıştır. Değiştirilen parametre değerleri ile bazı veri seti özelliklerinin daha iyi performans gösterdiğini belirlemişlerdir.

Bu tez çalışmasında, diğer çalışmalardan farklı olarak hem gerçek veri seti sayısı ve kullanılan sınıflandırma yöntemi sayısı arttırılmış hem de sınıflandırma

yöntemlerinin performanslarını karşılaştırmak amacıyla bir simulasyon çalışması gerçekleştirilmiştir.

2. VERİ MADENCİLİĞİ

Bu bölümde veri madenciliğine, kullanım alanlarına ve veri madenciliği yöntemlerine değinilmiştir.

2.1. Veri Madenciliği Nedir?

Veri madenciliği, büyük miktarda veriye sahip, gürültülü, bulanık uygulama verilerinin içine gizlenmiş, bilinmeyen ancak potansiyel olarak faydalı olabilecek bilgilerin açığa çıkarılması işlemidir (Shi, 2014). Adını, değerli cevherlerin yer altından çıkarılması gibi değerli bilgilerin veriden elde edilmesi sürecinden almaktadır. Veri madenciliği, veri tabanlarından verilerin edinildiği, istatistiksel modeller ve makine öğrenmesi ile bunlardan bilgi elde edilmesi süreci ile farklı alanların birleşiminden oluşmaktadır (Bhargava v.d., 2013). Veri madenciliği ile ilgili verilmiş bazı tanımlar şu şekildedir.

Ahmed (2004) veri madenciliğini, farklı kriterler kullanarak gömülü ve önceden bilinmeyen bilgileri, büyük veri tabanlarından çıkarmak için kullanılan ve kalıpları, ilişkileri keşfetmeyi mümkün kılan bir dizi teknik olarak tanımlamıştır.

Koh ve Tan (2011) veri madenciliğini, veri tabanları içerisinde önceden bilinmeyen kalıpları ve eğilimleri ortaya çıkarma, bu bilgiler ile tahmine dayalı modeller oluşturma süreci olarak tanımlarlar.

Gullo (2015) veri madenciliğini, büyük miktarda veriyi analiz ederek içerisinde bulunan kalıpları ve faydalı bilgileri ortaya çıkarma süreci olarak tanımlar.

Veri madenciliği ile ilgili tanımlamalar benzerlik göstermektedir. Bu tanımlamalardan yola çıkarak veri madenciliğinin kısaca, içerisinde büyük veri, istatistik ve makine öğrenmesini barındıran, verilerden yeni bilgileri çıkarmaya yarayan bir yöntem olduğunu söylemek mümkündür.

2.2. Veri Madenciliği Kullanım Alanları

Veri madenciliği, astronomi, sağlık, müşteri ilişkileri yönetimi, web uygulamaları, ağ, güvenlik, davranış ekolojisi ve iklim modellemesi gibi birçok alanda yaygın olarak kullanılmaktadır (Venkatadri v.d., 2011). Bu alanlardan elde edilen bilgiler yeni keşfedilen bir uzay cisminin gezegen veya yıldız olarak sınıflandırılması, hava durumu tahmini ve süpermarket reyonlarında ürün dizilim şekillerinin belirlenmesi gibi süreçler için kullanılabilir. Yani veri madenciliği, günlük hayatın kolaylaştırılmasından, bilime ve sağlığa kadar çok farklı alanlara yardımcı olmaktadır. Veri madenciliğinin kullanım alanlarına bazı örnekler şu şekilde verilebilir:

Pazarlama alanında pazar araştırması için kullanılmaktadır. Bu sayede müşterilerin hangi ürünleri birlikte satın alma eğiliminde olduğunu anlayıp öngörü yapabilmek ve satıcının buna göre karar vermesini sağlamak, satış rakamlarını arttırmak için kullanılır (Kaur ve Kang, 2016). Müşterilerin cevapladıkları anketler, kredi kartı hareketleri ve üyelik kartları gibi ortamlardan edinilen veriler ile ürün satışları arasında bağıntı kurularak birlikte satılan ürünler belirlenir. Ardından e-ticaret sektöründe "İlginizi çekebilecek ürünler" veya "Bu ürünü alanlar şunları da aldı" başlığıyla öneride bulunulur. Eğer ticaret sektörü web üzerinden değil ve bir market üzerinden ise birlikte satılması muhtemel ürünler yakın raflara yerleştirilerek satış değerlerinde artış yakalamak amaçlanır.

Bankacılık sektöründe dolandırıcılık tespiti, müşterilere kredi verilip verilmemesi veya kredi kart limitlerinin arttırılıp arttırılmaması, dolandırıcılık tespiti, müşteriyi kazanma, değerini arttırma ve müşteriyi elde tutmak için kullanılır (Basha, 2017). Bu sayede banka, kullanıcıları için uygulayacakları stratejileri geliştirebilmektedir. Aynı zamanda hisse senedi ve fon tahminlemesi için veri madenciliğinden yararlanılabilmektedir.

Tıp ve biyoloji alanında hastalık risklerini değerlendirmek, klinik kararlarını belirlemek ve desteklemek, hastalıkların gelişimini tahmin etmek, ilaçların etkilerini tahmin etmek için kullanılır (Zhang vd., 2016). Kan testleri ve sağlık taramalarından elde edilen veriler, çeşitli kanserlerin tanısında kullanılabilmektedir. Aynı zamanda

kalp verileri de kullanılarak kalp krizi riskinin tespiti, acil servislerde hastaların belirtilerine göre risk ve önceliklerin tespiti gibi alanlarda kullanılabilmektedir. Veri madenciliği başarılı tedavi sonuçları almak için etken olabilecek faktörlerin belirlenmesi, tedavi yöntemi geliştirilmesi, yanlış tedavi ve tanının tespiti gibi faydalarla tıp alanına yardımcı olmaktadır.

2.3. Veri Madenciliği Yazılımları

Büyük veriye olan ilginin artmasıyla birlikte bu verilerden elde edilebilecek alt bilgilerin değeri ve dolayısıyla bu amaçta kullanılabilecek araçlara olan ihtiyaç da artış göstermiştir. Veri madenciliği ve verilerin işlenmesi süreçlerinin gerçekleştirilmesi amacıyla farklı algoritmalar ve bu algoritmaları kullanan, makine öğrenmesine dayalı çalışan araçlar geliştirilmiştir.

Veri madenciliği için geliştirilen uygulamalar içerisinde açık kaynak kodlu bir yazılım olması, kullanım kolaylığı ve kullanıcı dostu bir arayüze sahip olması sebebiyle WEKA dikkat çekmektedir. WEKA, kendisiyle çalışacak olan veri madencilerine yedi farklı başlık altında yetmişten fazla sınıflandırma algoritması sunmaktadır.

Bu yazılım ilk olarak Yeni Zelanda'da bulunan Waikato Üniversitesi'nde Java programlama dili ile geliştirildi (Naik v.d., 2016). Java programlama dili ile yazılmış olması, WEKA'nın farklı sistemler üzerinde başarıyla çalışmasını sağladığı için geniş kitlelere ulaşmasına olanak tanımıştır.

WEKA, farklı uzantılara sahip dosya formatlarından verileri okuyabilmektedir. Bunların başında WEKA'da çalışılmış işlemlerin de kaydedildiği format olan .arff vardır. Bunun haricinde .csv, .data, .names gibi uzantılar olmakla birlikte kullanımında en çok tercih edilenler ve veri setlerinin tedarik edilişi sürecinde sıklıkla karşılaşılanlar .arff ve .csv'dir.

Veri madenciliği için kullanılan araçlardan bir diğeri de R'dır. Kütüphanesinde veri işleme ve görselleştirme için temel araçları barındıran ve istatistiki analizler için kullanışlı olan R, üçüncü kişilerin de kütüphaneye katkıda bulunabilmesini sağlar ve

bu amaçla kullanılabilecek paketlerin sayısı çarpıcı şekilde artmaktadır (Patil, 2016). Kullanıcılarına eşzamanlı olarak kodlama yapabilmeleri, aktif edilen kodların çıktılarını görmelerini, çağrılan veri setine ait değerleri görebilmelerini ve görselleştirme sonuçlarını sunabilen R-Studio, bu yönüyle çok kullanışlıdır. Excel ve SPSS gibi veri tabanlarından da veri elde edebilen R, farklı işletim sistemlerinde kullanılabildiği gibi 32 ve 64 bitlik sürümleri mevcuttur (Kalpana, 2017).

Veri madenciliği için kullanıcılara ücretsiz sunulan bir diğer araç Orange'dır. Yapılacak analiz ve görselleştirme işlemleri için widget denilen araçları kullanmaktadır. İçerisinde barındırdığı veri analiz ve veri görselleştirme araçlarına ek olarak Python yazılım dili ile eklemeler yapılabilen Orange, kullanıcıların widget'ları kullanarak bir iş akışı modelleyebilmeleri için büyük bir kullanım ekranı sağlar (Amala, 2019). Oluşturulan iş akış şemaları genellikle bir kaynaktan veriyi okuyan bir widget ile başlar ve bu veri üzerinde yapılmak istenen işleme ait diğer bir widget ile arasında bağ oluşturularak akış şeması oluşturulur.

Arayüz olarak geniş bir çalışma ekranı sağlayan diğer araç da KNIME'dır. Orange'da kullanılan widget'lar gibi KNIME da verileri modelleyen, veri dönüşümünü sağlayan, görselleştiren, sütun ve satır filtreleyen, verinin eğitim ve test verilerine bölünmesini sağlayan düğümler mevcuttur (Berthold v.d., 2009).

Bir diğer ücretsiz veri madenciliği ise RapidMiner'dır. 2001 yılında Dortmund Teknik Üniversitesi yapay zekâ biriminde geliştirilen yazılım, 2013 ve 2014 yıllarında yapılan yazılım anketleri doğrultusunda KDnuggets tarafından "en popüler veri analitiği yazılımı" olarak nitelendirilmiştir (Dwivedi vd. 2016). Bu veri madenciliği programının ücretsiz sürümü kullanıcılara açık olduğu gibi, lisanslandırma ile ücretli olarak PRO sürüm kullanmaya da izin vermektedir. RapidMiner da KNIME gibi boş bir kullanıcı arayüzü ekranında widget'ların birbirine eklendiği bir iş akış şeması oluşturularak çalışmaktadır. RapidMiner bünyesinde sınıflandırma, regresyon ve kümeleme amacıyla kullanılabilecek 100'den fazla öğrenme şeması barındırır.

2.4. Veri Madenciliği Yöntemleri

Yazılımlar sayesinde uygulanan veri madenciliğinde makine öğrenmesi önemli bir yer tutmaktadır. Makine öğrenmesi ve dolayısıyla veri madenciliği sürecinde kullanılan yöntemler, bu yöntemlerin veri kümelerinin sınıf bilgilerini kullanıp kullanmamasına göre denetimli ve denetimsiz yöntemler olarak ikiye ayrılmaktadır (Dash v.d., 2011).

Denetimli makine öğrenme yöntemleri, daha önce görülmemiş, yeni veriler ile ilgili sınıf değerlerinin tahmin edilmesi için eğitim verilerinden faydalanan bir makine öğrenmesi yöntemidir (Aggarwal v.d., 2012). Sınıf veya hedef etiketine sahip veri setleri ile kullanılabilen bu yöntemde kullanıcı, yeni oluşturulan bir gözlem verisine ait elde etmek istediği sınıf değerini tahmin edebilmek için öncelikle makineye bir eğitim seti sunar. Ardından, makine bu süreçten edindiği bilgilerle bir model oluşturur ve bu model ile yeni veri için sınıf değerini tahmin ederek bilgiyi kullanıcıya sunar.

Sınıf etiketine sahip olan veriler ile denetimli makine öğrenmesinde kullanılabilecek farklı algoritmalar mevcuttur. Rashidi vd. (2019) bu algoritmaların lineer regresyon, lojistic regresyon, k-en yakın komşuluklar, Naive Bayes, DVM, yapay sinir ağları ve rasgele orman olduğundan bahseder.

Denetimsiz makine öğrenmesi yöntemleri sayısal öznitelikleri normalleştirme veya standartlaştırma, öznitelik değerlerini birleştirme, özellikleri kaldırma, dize özelliklerini nominal değerlere dönüştürme, zaman serisi verilerini işleme koyma gibi işlemler yapabilir ve seyrek örnekleri, seyrek olmayan örneklere dönüştürerek örnekleri belirli ölçütlere göre kaldırır (Frank v.d., 2010). Amaç, bir çıktı(sınıf) değişkeninin olmadığı veri seti tiplerinde veriler hakkında daha fazla bilgi edinmek için temel yapıyı veya dağılımı modellemektir. Veri setleri içerisinde bir çıktı bilgisi bulunmadığı için doğru bir cevap veya tahminleme yoktur. Verilerdeki yapıyı keşfetmek ve ortaya çıkarmak için algoritmalar kendi tasarımlarını kullanırlar.

Denetimsiz makine öğrenmesi yöntemlerinden birisi kümelemedir. Kümeleme analizinde amaç, veri setinin kendi özelliği ve gözlemlerine göre sayısı önceden belirlenebilen gruplara ayırmaktır (Chen, 2020).

Veri madenciliği sürecinde kullanılan yöntemler, denetimli ve denetimsiz yöntemler ana başlığı altında sınıflandırma, kümeleme ve birliktelik analizidir. Hangi veri madenciliği yönteminin kullanıldığına bakılmaksızın, ilk olarak veri önişleme sürecinin işlenmesi gerekmektedir. Bir sonraki bölümde veri önişleme ile ilgili kısa bir bilgi sunulmaktadır.

2.4.1. Veri önişleme

Bir veri setinden elde edilebilecek en doğru alt bilgileri elde etmek veri madenciliğinin amaçlarındandır. Bu yüzden kullanıcının veriyi tanıması, gerekli durumlarda düzenlemesi de doğru veri madenciliği yöntemini kullanması kadar önemlidir. Veriye erişimin ve saklanmasının kolay olması, beraberinde veri kirliliğini de getirmektedir.

Elde edilen veriler ham halleriyle gürültülü, tutarsız ve/veya eksik gözlem içerebilmektedirler. Veri setinin bu hâliyle işleme sokulması, veri madenciliği yönteminden elde edilecek başarıyı ve bilgiyi önemli ölçüde düşürerek kullanıcıyı yanıltabilmektedir.

Veri önişleme, temel olarak birbiriyle bağlantılı olan veri temizleme, veri birleştirme, veri dönüştürme ve veri azaltma işlemleriyle dört adımda yapılmaktadır (Al-Taie v.d., 2019).

Veri temizleme:

Ham veri setleri içerisinde, veri setini tutarsız ve düşük kaliteli kılan eksik gözlem, aykırı değerler, anlamsız değişkenler, insan veya ölçüm aletlerinin hatalarından kaynaklı yanlış girilmiş gözlemler içerebildiği için bu veri setlerinden elde edilecek sonuçlar da kaliteli olmayacaktır (Jony v.d., 2015). Kullanıcı, madencilik süreci sonunda gerçek ve anlamlı sonuçlar elde edebilmek için veri setinin kalitesini arttırmalıdır. Bu amaç doğrultusunda veri seti içerisinden aykırı ve tekrarlayan gözlemler ile birlikte gereksiz değişkenler çıkarılmalı, eksik gözlemler tahmin edilerek yenilenmeli veya veri setinden çıkarılmalıdır.

Veri birleştirme:

Bazı veri setlerinde birbiri ile eş anlam ifade edebilecek değişkenler farklı kolonlarda işlenmiş olabilmektedir. Bu durum makine öğrenmesi sürecinde gereksiz bilgi fazlalığı yaratacağından model optimal kurulamayabilir ve alınan çıktı gerçeği yansıtmayabilir. Modelden en iyi sonucu alabilmek için makine öğrenmesi sürecinde tüm gerekli bilgiler, en az karmaşıklık ile verilmelidir. Bu sebeple iki değişkenin tek bir kolonda verilmesi sağlanmalıdır. Aynı zamanda farklı veri bankalarından aynı veri tipine ait verilerin birleştirilmesi süreci de bu sürecin parçasıdır ve hatalı yapılan bir birleştirme işlemi, veri setinin anlam bütünlüğünü bozacağı için bu adım dikkatli bir şekilde yapılmalıdır.

Veri dönüştürme:

Veri setleri içerisindeki değişkenler ve bunlara ilişkin gözlemler çeşitlilik gösterebilmektedir. Bir veri seti numerik ve kategorik verileri bir arada içerebilir. Kimi durumlarda ise makine öğrenmesi sürecinde kategorik veriler numerik, numerik veriler ise kategorik olarak algılanabilmektedir veya değişkene ait numerik gözlemler çok geniş bir dağılıma sahip olabilmektedir. Bu durumda madencilik süreci uzun sürebilmekte, hatta gerçeği yansıtmayan sonuçlar elde edilebilmektedir. Bu sorunların önüne geçebilmek amacıyla kullanıcı verilerin kategorik ve numerik arasındaki dönüşümünü sağlama, verileri normalleştirme gibi işlemlerle veri setini makine öğrenmesi ve madencilik süreci için uygun hâle getirmelidir(García v.d., 2015).

Kullanıcı, veri setine ait gözlem değerlerini normalleştirmeyi seçebilir. Normalleştirme işleminde amaç, gözlem değerleri içerisinde varsa aykırı değerlerin etkisinin azaltılması, en büyük ve en küçük gözlemler arasındaki boşluğun daraltılması ve bu sayede sınıflandırma başarısında artış sağlamaktır. Bu amaçla kullanımına başvurulan yöntemlerden bazıları aşağıdaki gibidir:

Z-Skoru Yöntemi:

Bu yöntem ile veri seti içerisindeki değerler 0 ortalama değerine ve 1 standart sapma değerine sahip olacak şekilde ölçeklenirler (Cao v.d.2016). Z-skoru normalleştirme sürecinde kullanılan yöntem, $x_i^!$ normalize edilmiş gözlem, x_i gözlem değeri, μ_X

özniteliğin aritmetik ortalaması, σ_X özniteliğin standart sapması olmak üzere aşağıdaki gibidir:

$$x_i^1 = \frac{x_i - \mu_X}{\sigma_X} \quad (2.1)$$

Min-Maks Yöntemi:

Verilerin normalleştirilmesi için kullanılan bir diğer yöntem min-maks yöntemidir. Bu yöntem sonucunda gözlemler 0 ve 1 arasında değerlenir, aykırı gözlemlerin etkisi azaltılmış olur ve daha küçük standart sapma sonuçları elde edilir (Abdeldaim v.d., 2018). Min-maks yöntemine ait denklem, x_i^1 normalleştirilmiş gözlem değerini, x_i orijinal gözlem değerini, x_{min} özniteliğe ait en küçük gözlem değerini ve x_{max} özniteliğe ait en büyük gözlem değerini belirtmek üzere aşağıdaki gibidir:

$$x_i^1 = \frac{x_i - x_{min}}{x_{max}} \quad (2.2)$$

Veri azaltma:

Veri seti hacmi ile makine öğrenmesi sürecinde harcanacak zaman doğru orantılıdır. Bu yüzden çok büyük hacimli veri setlerinde, eğitim verisi ile çalışılıp bir model oluşturulması önemli ölçüde zaman almaktadır. Olağandan çok yüksek olan gözlem sayılarına sahip veri setleri ile çalışırken veri setinden bir örneklem çekilmesi, boyut küçültmek için yararlı olmakla birlikte indirgenmiş örneklem daha az gürültü içerebileceği için oluşturulacak modelin başarısını da arttırabilmektedir (Islam v.d., 2010).

Veri azaltma amacıyla veri önişleme sürecinde kullanılan yöntemler aşağıdaki gibi sınıflandırılabilir:

1-Veri birleştirme(Veri küpü)

2-Boyut indirgeme

3-Veri sıkıştırma

4-Kesikli hale getirme

2.4.2. Sınıflandırma

Farklı değişken özelliklerine ve nihayetinde bir sınıf değişkenine sahip veri setlerinde, veri setine yeni dâhil olacak bir gözlemin hangi sınıfa atanacağının tahmin edilmesini amaçlar. Bu süreçte elde bulunan verilerden bir model oluşturulur ve yeni gözlemin sınıf değeri bu model ile tahmin edilir. Farklı veri tipleri için farklı tahmin ediciler geliştirilmiştir.

2.4.3. Kümeleme

Veri setindeki gözlemlerin birbirlerine benzerlik ve farklılıklarının incelenerek alt kümelere ayrılması sürecidir. Bu sayede büyük veri setlerini okumak kolaylaşır ve yeni anlamlar çıkarılmasına yardımcı olur.

2.4.4. Birliktelik kuralları

Denetimsiz bir makine öğrenmesi yöntemi olan birliktelik kuralları madenciliği, büyük veriler içerisindeki kümeler arasındaki bağlantı ve ilişkileri ortaya çıkarır. Bu kural, bir öğe kümesinin bir işlemde ne sıklıkla meydana geldiğini göstermektedir.

Özellikle süpermarketler tarafından kullanılan bu yöntemde amaç birlikte satın alınma eğiliminde olan ürünleri tespit etmek ve müşterileri tanımaktır. Bu yöntemde her alışveriş için satın alınan ürünlerin bilgisi depolanır ve bu veriler üzerinden birlikte satın alınmaya en yatkın olan ürünler belirlenir, bu sayede reyonlarda bu ürünler müşterinin göz hizasında ve birbirine yakın konumlandırılarak satışların arttırılması planlanır.

3. SINIFLANDIRMA

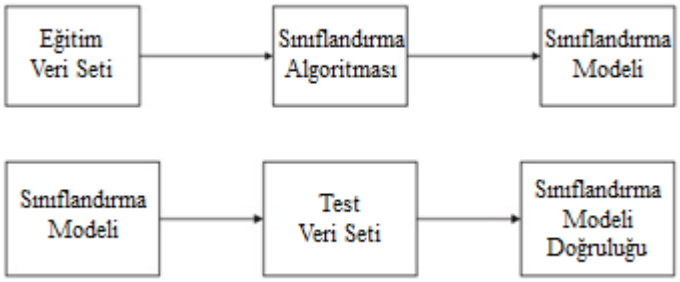
Makine öğrenmesi ve istatistik temelli bir metot olan sınıflandırma, denetimli bir veri madenciliği tekniğidir. Gözlem değerlerine bağımlı olarak bir sınıf değişkenine sahip veri setlerinde kullanılabilir. Bu veri setlerinde her bir veri, içerdiği gözlemlerin değerlerine bağlı olarak bir sınıf değerine sahiptir. En az iki adet sınıf değerine sahip olurlar ancak veri setinin büyüklüğü ve gözlemlerin açıklayabildikleri değişkenlere göre bu sayı çok büyük değerlere ulaşabilmektedir. Veri seti içerisinde sınıf değerleri arasında sayıca en çok örneğe sahip sınıfa majör sınıf, en az örneğe sahip olan sınıfa ise minör sınıf denilmektedir ve algoritmaların çoğu sınıflandırma sürecinde majör sınıfa yoğunlaşır(Longadge v.d., 2013).

Sınıflandırmanın amacı bir veri setindeki gözlemlerin sınıf değişkenine ait bilinen değerlerinin ele alınması ve sınıf değişkeni bilinmeyen yeni bir gözlem sunulduğunda, sınıflara ait olduğu bilinen verilerin karakteristiklerini inceleyerek edindiği bilgiyi kullanarak sınıf değeri tahmin etmektir (Pandey v.d., 2011).

Veri madenciliğinde sınıflandırma süreci, makine öğrenmesine dayalı olarak, bir eğitim veri seti kullanılarak model oluşturulması ve bir test veri seti ile bu modelin başarısının test edilmesi şeklinde iki adımdan oluşmaktadır (Soiraya v.d., 2012). Öğrenme adımı olarak adlandırılan ilk adımında, sınıflandırma algoritmasının eğitilmesi için, veri setindeki gözlemlerden faydalanılır ve bir model oluşturulur. Algoritmanın eğitilmesi için kullanılan bu verilere eğitim verisi adı verilir. Eğitilmiş olan algoritmaya ait bu model, test verileri ve yeni veriler için tahmin yapmakta kullanılabilir. Elde edilen eğitilmiş modelin sınıf tahmini yapma konusundaki başarısının test edilmesi için veri seti içerisinde eğitim için kullanılmamış veriler kullanılır. Bu verilere test verisi adı verilir. Test verisi olarak eğitim verisine ait gözlemler kullanılması, sınıflandırmanın başarısını önyargılı bir şekilde etkileyecektir. Eğitim ve test işlemi için verinin iki ayrık alt kümeye bölünmesi gerekmektedir. Veri setinin bölünmesi işlemi sırasında verinin ne kadarının eğitim ve ne kadarının test için ayrılacağının belirlenmesi gerekmektedir. Fazla büyük bir eğitim verisi ile çalışmak, test için az veri bırakacak ve yapılan test sonucu elde edilen başarı düzeyi gerçeği yansıtmayacaktır. Diğer yandan test verisinin yüksek

boyutlarda olması ve eğitim için küçük bir veri kümesi kalması ise makine öğrenmesi sürecini aksatacak ve başarılı bir model kurulamadığı için başarı düzeyi düşük çıkacaktır.

Sınıflandırma sürecine ait temel adımların işleyişleri şekil 3.1.'de görülmektedir.



Şekil 3.1. Sınıflandırma sürecinin adımlarının işleyişi (Huang ve Liang, 2019)

Sınıflandırma algoritmasının eğitilmesi için test amacıyla kullanılacak verinin seçilmesi için farklı yöntemler bulunmaktadır, bu yöntemler bir sonraki bölümde verilmektedir.

Günümüzde pek çok farklı alanda veriler oluşturulmakta ve saklanmaktadır. Sınıflandırma için kullanılabilecek veri setlerinin sahip oldukları gözlemlere ait veri tipleri de değişiklik göstermektedir. Tüm gözlemler kategorik olabildiği gibi kategorik ve numerik gözlemlerin karışık olduğu veri setleri ile de karşılaşabilmekteyiz. Aynı zamanda sınıf sayıları da çeşitlilik göstermektedir. Veri setlerindeki bu çeşitlilik, zaman içerisinde farklı sınıflandırma algoritmalarının geliştirilmesine sebep olmuştur. Algoritmaların çeşitli gereksinimleri olabilmektedir. Örneğin kimi algoritmalar yalnızca iki adet sınıf değerine sahip veri setleri ile çalışırken, kimi algoritmalar sürekli veriler ile çalışmamaktadır.

3.1. Test Seti Seçim Yöntemleri

Bu bölümde sınıflandırma algoritmalarının, veri seti gözlemleri ile eğitilmesi için geliştirilmiş farklı yöntemlere değinilecektir.

3.1.1. Hold-out yöntemi

Hold-Out yönteminde, veri setinin belli bir yüzdesi kadarı eğitim verisi, kalan yüzdesi kadarı ise test verisi olarak ayrılır (Omary ve Mtenzi, 2009). Makine öğrenmesi eğitim verisi ile gerçekleşir ve bu veriler üzerinden model oluşturulur. Ardından modelin başarısı test verisi ile test edilir. Yöntemin dezavantajlarından birisi, iki alt grubun aynı olmaması durumunda, yani test verilerinin eğitim verilerini temsil edemediği durumda modelin başarısız sonuçlar vermesidir. Örneğin sınıflandırma işlemi veri setindeki gözlemlerin sınıf değerlerine göre sıralanmasının ardından yapılırsa test verisi içerisinde, eğitim verilerinde görülmemiş sınıf değerleri olacak ve başarı büyük ölçüde düşecektir. Hold-Out yönteminin bir diğer dezavantajı ise büyük veri setlerini anlamak konusunda başarılı bir yöntem olmasına karşın az gözleme sahip veri setleri için zayıf kalmasıdır. Çünkü küçük veri setlerini ikiye ayırmak çok daha küçük verilerle çalışmak anlamına gelmektedir. Eğitim için kullanılan veriler veri setini tam anlamıyla temsil edemeyeceği için Hold-Out yöntemi büyük veri setleri için daha uygundur. Bu tezde yürütülen makine öğrenmesi süreçlerinde Hold-Out yöntemi kullanılmıştır.

3.1.2. Çapraz doğrulama yöntemi

Az gözlem sayısına sahip veri setleri ile çalışmalarda tutarlılığı en yüksek olan bu yöntemde ilk olarak veri seti k adet parçaya bölünür. Bu k adet parçadan bir tanesi test verisi, kalan parçalar eğitim verisi olacak şekilde kullanılır ve bir başarı değeri elde edilir. Bu işlem her parça test verisi olarak kullanılacak şekilde devam eder ve sonunda k adet başarı değeri bulunur. Elde edilen bu k adet başarı değerinin aritmetik ortalaması ise sınıflandırma algoritmasının nihai başarısı olarak sunulur. Veri setinin gözlem sayısı düşük olsa dahi her gözlemi eğitimde kullanması çapraz doğrulama yönteminin avantajıdır, sınıflandırma algoritmasının nihai başarısını sunmak için k defa sınıflandırma yapması ise süreyi uzattığı için dezavantajıdır (Olson ve Delen, 2008).

3.1.3.Tek çıkış yöntemi

Bu yöntem, çapraz doğrulama yönteminin özel bir hâlidir. N gözleme sahip bir veri setinde bir gözlem test verisi olarak ayrılır ve geri kalan N-1 gözlem eğitim verisi olarak kullanılır. Bu işlem veri setindeki her gözlem test verisi olarak kullanılana kadar devam eder. Yani tek çıkış yöntemi, çapraz doğrulama yönteminin k değeri için gözlem sayısı olan N değerinin girilmesi ile oluşan türüdür. Dolayısıyla sınıflandırma işlemi N kez tekrarlanmış olur ve her sınıflandırmadan elde edilen başarı değerlerinin aritmetik ortalaması nihai başarıyı verecektir. Veri seti içerisindeki her gözlemin model eğitilirken kullanılması bu yöntemin en büyük avantajıdır, sınıflandırma işleminin N kez tekrarlanması ise yüksek maliyet ve zaman kaybına sebep olacağı için dezavantajıdır.

3.2. Sınıflandırma Başarısı Ölçme Yöntemleri

Veri setinden bir sınıflandırıcı ve test seti seçme yöntemi yardımıyla bir model oluşturulup, sınıf tahminleri yapılmasından sonra izlenmesi gereken adım, modelin ne kadar etkili olduğunu incelemektir. Model performanslarını ölçmek için farklı ölçütler mevcuttur. Bu ölçütlerden en çok kullanılanları aşağıdadır.

3.2.1. Doğru sınıflandırılmış örnek oranı

Veri madenciliği yöntemlerinden sınıflandırmanın temel amacı olabilecek en kısa sürede en fazla gözlemin sınıflarının doğru şekilde tahmin edilmesini sağlamaktır. Doğru sınıflandırılmış örnek oranı(DSÖO) değerinin hesaplanmasında kullanılan formül şu şekildedir:

$$\text{DSÖO} = \left(\frac{\text{Doğru Sınıflandırılmış Örnek Sayısı}}{\text{Toplam Örnek Sayısı}}\right) \tag{3.1}$$

Bu ölçüt, sınıflandırmanın performansını ölçmek için kullanılan en basit ve en çok kullanılan yöntemdir (Ferri v.d., 2009). Yorumlanması kolay olmasına karşın yüksek

doğruluk oranı o sınıflandırıcının mutlak iyi bir başarı sergilediğini göstermez. Örneğin yapılan yanlış sınıf tahminlerinin hangi sınıfta yoğunlaştığı bilgisini bize vermez.

3.2.2. Duyarlılık(geri çağırma) ve özgünlük

Duyarlılık ve özgünlük ölçütlerini anlamak için iki adet sınıf değerine sahip bir veri seti ile yapılmış sınıflandırma örneği için karışıklık matrisi belirtelim.

Çizelge 3.1. Örnek karışıklık matrisi

		Gerçek	
		Pozitif	Negatif
Tahmin	Pozitif	DP	YP
	Negatif	YN	DN

Verilen karışıklık matrisi içerisinde bulunan DP değeri, veri seti içerisinde sınıf değeri pozitif iken sınıflandırma sonucunda pozitif olarak tahmin edilen gözlemlerin, YP ise sınıf değeri pozitif iken negatif olarak tahmin edilen gözlemlerin sayısını temsil etmektedir. YN değeri, sınıf değeri negatif iken pozitif olarak tahmin edilen, DN ise sınıf değeri ise sınıf değeri negatif iken negatif olarak tahmin edilen gözlemlerin sayısını temsil etmektedir.

Duyarlılık ölçütü, sınıflandırma sonucunda sınıf değeri pozitif olanlar içerisinde pozitif olarak tahmin edilen gözlemlerin oranıdır ve aşağıdaki şekilde formüle edilir:

$$\text{Duyarlılık} = \left(\frac{\text{DP}}{\text{DP+YN}}\right) \tag{3.2}$$

Özgünlük ölçütü ise sınıf değeri negatif olan gözlemler içerisinde sınıflandırma sonucunda negatif olarak tahmin edenlerin oranıdır ve aşağıdaki şekilde formüle edilir:

$$\text{Özgünlük} = \left(\frac{\text{DN}}{\text{DN+YP}}\right) \tag{3.3}$$

3.2.3. Kesinlik

Sınıflandırma sonucunda sınıf değeri doğru şekilde pozitif olarak tahmin edilmiş örnek sayısının, sınıfı pozitif olarak tahmin edilen tüm örneklerin sayısına oranıdır ve aşağıdaki şekilde formüle edilir:

$$\text{Kesinlik} = \left(\frac{\text{DP}}{\text{DP+YP}}\right) \tag{3.4}$$

3.2.4. F-ölçümü

Duyarlılık ve kesinlik ölçülerinin harmonik ortalaması ile hesaplanan F-ölçümü, sınıflandırıcıların performanslarını ölçmenin geleneksel bir yoludur ve aşağıdaki şekilde formüle edilir:

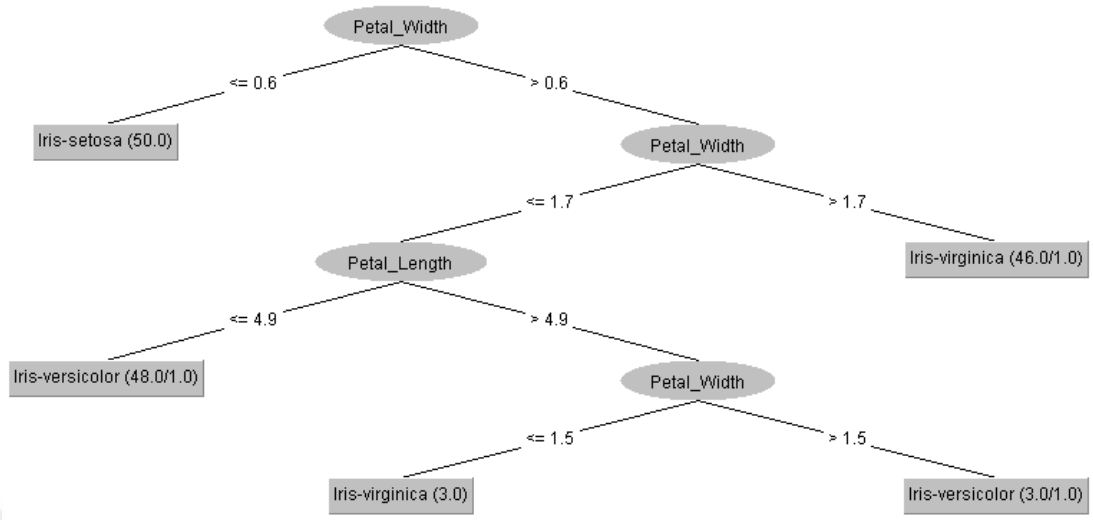
$$\text{F} - \text{Ölçümü} = \frac{2*Duyarlılık*Kesinlik}{Duyarlılık+Kesinlik} \tag{3.5}$$

3.3. Sınıflandırma Yöntemleri

Bu bölümde literatürde en çok kullanılan sınıflandırma yöntemlerine ve çalışma prensiplerine değinilecektir.

3.3.1. Karar ağaçları

Karar ağaçları bünyesindeki algoritmalar, verideki gözlemlere ait olan tüm eylemleri, bu eylemlerin olası seçeneklerini ve sınıf değerlerini şematik olarak sergileyen bir algoritmadır. Bu sebeple anlaşılması ve yorumlanması gayet kolaydır. Adını, şematik gösterimin karar ve yaprak düğümleri olan ağaç benzeri yapısından almaktadır. Ağacın en üstünde karar düğümü bulunur ve dallara ayrılır. Ağaç dallanmalarının sonucunda, sınıf değerini temsil eden bir yaprak düğümüne bağlanmaktadır. Şekil 3.2.'de Iris veri seti ile oluşturulmuş bir karar ağacı şeması gösterilmektedir.



Şekil 3.2. Iris veri seti karar ağacı şeması

Karar ağaçlarının makine öğrenmesi sürecinde ortaya koydukları modelleri yorumlamak kolaydır ve eğitim süreci uzun sürmediği için büyük gözlem değerlerine sahip veri setlerinde veri madenciliğinin uygulanması daha az zaman alacaktır (Zhao v.d., 2008). Kök düğümünden başlayarak dallara, oradan da nihayetinde yaprak düğümlerinde bulunan sınıf değerlerine uzanan bu karar ağacı yapısını kullanan farklı algoritmalar bulunmaktadır.

3.3.1.1. Aşırı öğrenme

Bir sınıflandırma algoritması ile çalışırken, eğitim verileri ile modelin eğitilmesi sınıflandırmanın başarısını paralel bir şekilde etkilemektedir. Ancak karar ağaçları sınıflandırıcılarında, ağaç modeli eğitim verileri tarafından mükemmel şekilde eğitilebilmektedir. Eğitim verileri ile bu kadar iyi bir şekilde eğitilen model, test verilerini veya sonradan eklenen gözlemleri tanımakta zorluk yaşayabilmekte, doğru sınıflandırma performansı düşmektedir. Bu durum aşırı öğrenme problemi olarak adlandırılmaktadır (Kulkarni ve Shrestha, 2017).

3.3.1.2. Ağaç budama

Ağaç tabanlı sınıflandırma algoritmaları hem yorumlanması basit olması sebebiyle hem de genelde yüksek doğrulukta sınıflandırma yapması sebebiyle tercih edilmektedir. Ancak kimi zaman oluşturulan ağaç modeli, eğitim verisini mükemmel şekilde öğrenir ve model aşırı öğrenme problemi yaşanır. Bu durumda model, daha önce hiç karşılaşmadığı bir gözlemle karşılaştığında yanlış sınıflandırma yapma eğiliminde olur. Kimi zaman ise ağaç modeli çok geniş olabilir, sınıflandırma süreci fazla uzun sürebilir ve maliyete sebep olabilir. Bu gibi durumların önüne geçebilmek için modelde bulunan alt ağaçların budanması gerekmektedir. Böylece aşırı öğrenme durumunda sınıflandırma algoritmasının tahmin yapabilmesi sağlandığı gibi sınıflandırma için harcanan süre azaltılmış, dolayısıyla maliyet düşürülmüş olur.

Karar ağaçlarının budanması için özellikle kullanılan iki yöntem vardır. Bunlar ön budama ve sonradan budamadır. Özellikleri iyi bilinmeyen veri setleri için önce modelin eğitilmesi ve ağacın oluşturulması sağlandıktan sonra ağaç budaması yapılır ve buna sonradan budama denirken, özellikleri iyi bilinen bir veri seti için ağaç budama işlemi ağaç oluşturulurken yapılabilir ve bu işleme ön budama denir (Sim v.d., 2017).

3.3.1.3. J48

J48 algoritması, yine kendisi gibi karar ağacı temelli bir algoritma olan ID3 algoritmasının özelliklerine ek olarak eksik gözlemlerle çalışabilen, ağaç budaması yapabilen ve C4.5 algoritmasının WEKA veri madenciliği uygulaması için Java tabanlı, açık kaynak kodlu olarak düzenlenmiş hâli olan bir algoritmadır (Kaur v.d., 2014).

Ağacın oluşturulmasına öncelikle kök düğümden başlanmaktadır. Veri setindeki hangi değişkenin kök düğümüne yerleşeceğinin ve ardından dalların hangi sıra ile dizilecekleri, karar ağacı algoritmasının başarı oranını etkileyeceği için çok önemlidir. Algoritma, kök ve dal düğümlerine optimal değişkenleri atayabilmek amacıyla değişkenlerin kararsızlık derecelerini ölçen entropi kullanılır (Bienvenido-

Huertas v.d., 2019). Entropi değerinin hesaplanmasında aşağıdaki formül kullanılmaktadır:

$$E(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{3.6}$$

Entropi değerinden yararlanılarak elde edilen bilgi kazancı formülü ise şöyledir:

$$IG(S, A) = E(S) - \sum_{v \epsilon V(A)}^{c} \frac{S_V}{S} E(S_v) \tag{3.7}$$

Entropi ve bilgi kazancına ait formüller (3.6) ve (3.7) içerisinde bulunan S, örnekleri göstermektedir. c, sınıf değişkenine ait sınıf sayısını, V(A), A değişkenini içerir değerler kümesini, S_V ise A değişkeni içerisinde V değerine sahip gözlemleri temsil etmektedir. (3.6) ve (3.7) den hareketle entropisi, yani karmaşıklığı en düşük olan değişkenin bilgi kazancının en büyük olacağı görülmektedir. Algoritma, ağacı oluşturmaya kök düğümünden başlamakta ve kök düğümüne, bilgi kazancı en yüksek değişkeni atamaktadır. Diğer değişkenler için yapraklara atamalar yapılması süreci yine bilgi kazancına dayalı olarak devam etmektedir. Örneklerin hepsinin aynı sınıfa ait olması, örnekleri bölecek özellik kalmamış olması ve kalan özelliklerin değerini taşıyan örnek bulunmaması durumlarında süreç sona erer.

3.3.1.4. Rasgele orman

Rasgele Orman algoritması, temelinde birçok karar ağacının birleşmesinden oluşan bir algoritmadır, karar ağaçları birleşerek ormanı oluşturur. Orman düzeni, veri seti içerisinde eğitim için ayrılan verilerden, kullanıcının da belirleyebileceği sayıda örneklem çekilip, bu örneklemlere ait karar ağaçları elde edilmesi ile oluşturulur (Everingham v.d., 2016).

Rastgele orman sınıflandırma algoritmasında dallardaki düğümlerin belirlenmesi sürecinde heterojenlik ölçüsü olan Gini İndeksi kullanılır. Gini indeksi, bir özelliğe ait değerlerin ortalamasının alınarak ortalamadan küçük olan gözlemlerin sol gruba, eşit ve büyük olanların ise sağ gruba alınması ile ikiye ayrılması ve bu iki gruba ait indeks değerlerinin hesaplanması ile gerçekleşir (Yılmaz, 2018).

$$\mathrm{Gini}_{\mathrm{Sağ}} = 1 - \sum_{i=1}^{n} \left(\frac{R_i}{|D_{\mathrm{Sağ}}|}\right)^2 \quad (3.8)$$

$$\mathrm{Gini}_{\mathrm{Sol}} = 1 - \sum_{i=1}^{n} \left(\frac{L_i}{|D_{\mathrm{Sol}}|}\right)^2 \quad (3.9)$$

Sağ ve sol gruplara ait Gini indeksi hesaplamalarında kullanılan (3.8) ve (3.9) formülleri içerisinde n sınıf sayısını, $D_{Sağ}$ özelliğe ait hesaplanan ortalamadan büyük veya eşit olan gözlem sayısını, D_{Sol} özelliğe ait hesaplanan ortalamadan küçük olan gözlem sayısını, R_i sağ grupta bulunan ve i sınıfına ait olan gözlem sayısını, L_i ise sol grupta bulunan i sınıfına ait gözlem sayısını ifade etmektedir. Her özelliğe ait Gini indeksi aşağıdaki formül ile hesaplanır:

$$\mathrm{Gini}_i = \frac{1}{k}\left(|D_{\mathrm{Sol}}|G_{\mathrm{Sol}} + |D_{\mathrm{Sağ}}|G_{\mathrm{Sağ}}\right) \quad (3.10)$$

Gini İndeksi hesaplamasında kullanılan (3.10) formülleri içerisinde bulunan k eğitim veri setine ait gözlem sayısını, $G_{Sağ}$ sağ gruba ait Gini İndeksi değerini, G_{Sol} sol gruba ait Gini İndeksi değerini belirtmektedir.

3.3.2. Bayes sınıflandırıcılar

Bu bölümde bulunan sınıflandırma algoritmaları, geçmişte meydana gelen olaylardan elde edilen bilgiler ile aynı olayın meydana gelme olasılığını hesaplamakta kullanılan Bayes Teoremi'ne dayanmaktadır. Bayes Teoremi aşağıdaki şekilde ifade edilmektedir:

$$P(B\backslash A) = \frac{P(A\backslash B)P(B)}{P(A)} \quad (3.11)$$

Bayes Teoremi'ni ifade eden (3.11)'de bulunan P(B\A) ifadesi A olayının gerçekleştiği durumda B olayının gerçekleşme olasılığını, P(A\B) ifadesi B olayının gerçekleştiği durumda A olayının gerçekleşme olasılığını, P(A) ifadesi A olayının gerçekleşme olasılığını, P(B) ifadesi ise B olayının gerçekleşme olasılığını ifade etmektedir.

Bayes sınıflandırma algoritmaları, eğitim sürecinde örneklerin hangi sınıfa, hangi olasılıklarla dâhil olduklarını öğrenir. Ardından test verisi içerisinden gelen bir gözlemdeki örnekleri inceleyerek, eğitim sürecinden elde ettiği olasılık değerleriyle karşılaştırmasını yapar ve en büyük olasılık değerine sahip olan sınıf için sınıf tahmini yapar.

3.3.1.1. Naive Bayes sınıflandırıcısı

Naive Bayes sınıflandırma algoritması, özelliklerin birbirinden bağımsız olduğu ve her özelliğin eşit önem derecesine sahip olduğu varsayımlarına dayanmaktadır (Wu v.d., 2017).

Naive Bayes sınıflandırma algoritması denklemi (3.12)'de bulunan ifadeler C, m sınıflı (C_1,C_2,…,C_m) kümedeki bir sınıfı ve X_k ise k özellikler vektöründen bir özelliği yani X_k=[x_1,x_2,…,x_k] olmak üzere revize edildiğinde Naive Bayes sınıflandırma algoritmasına ait sınıf tahminine ilişkin denklem aşağıdaki gibidir (Agraval v.d., 2008):

$$P(C_i \backslash X_k) = \frac{P(X_k \backslash C_i)P(C_i)}{P(X_k)} \quad i = 1,2,\dots,m \text{ için} \tag{3.12}$$

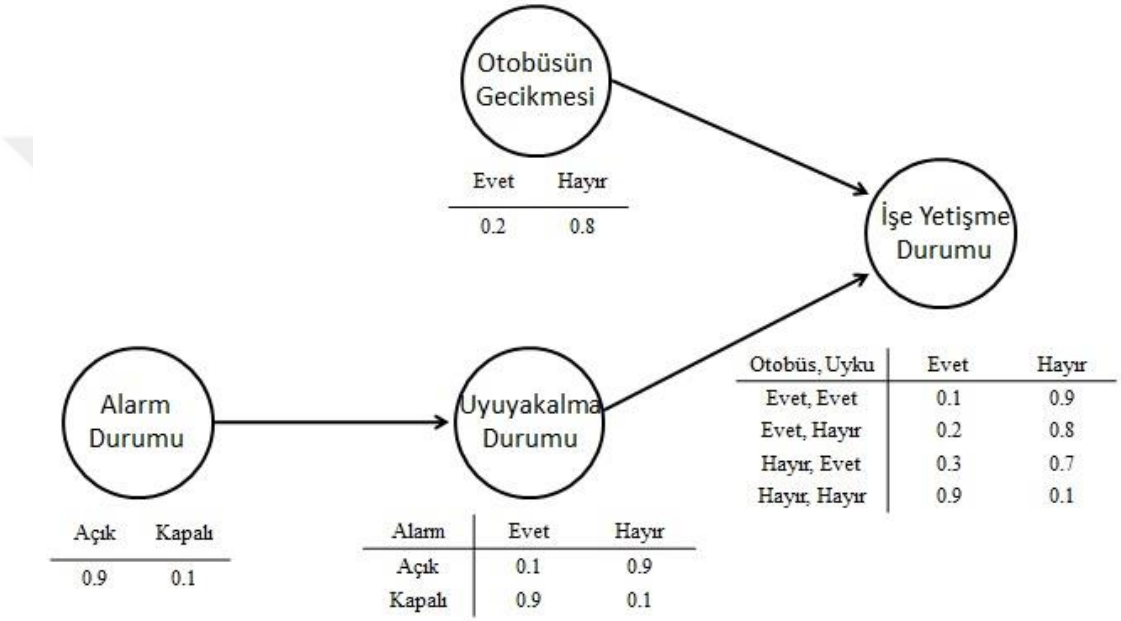
Naive Bayes sınıflandırma algoritmasına ait (3.12) içerisindeki $p(X_k|C_i)$, $p(C_i)$ ve $p(X_k)$ ifadeleri, yazılıma sunulan eğitim verisi kullanılarak hesaplanmaktadır. Ardından test verisi içerisindeki her gözlem için X_k sınıf tahmini, aşağıdaki fonksiyon ile hesaplanır.

$$f(X_k) = \text{argmax}\, p(C_i|X_k) \tag{3.13}$$

Naive Bayes algoritmasının çalışma prensibinin, eğitim verisi içerisindeki gözlemler içerisindeki her özelliğin hangi sınıfa ait olduğunu hesapladığı, sınıf değeri belli olmayan bir gözlemde ise özelliklerinin aitlik olasılıkları ile yaptığı hesap sonucunda olasılık değeri daha yüksek çıkan sınıfa atama yaptığı (3.12) ve (3.13) incelendiğinde söylenebilir.

3.3.1.2. Bayes ağları sınıflandırıcısı

Bayes Ağları sınıflandırma algoritmasında oluşturulan model, düğümler ve düğümleri birbirine bağlayan oklar ile biçimlendirilir. Bayes Ağları algoritmasında değişkenler düğümlerle gösterilir ve bu düğümler birbirine yönlü ve döngüsel olmayan bağlarla bağlıdır, bağlı olmayan düğümler ise aralarında koşulsuz bağımsızdır (Hamoud v.d., 2017).



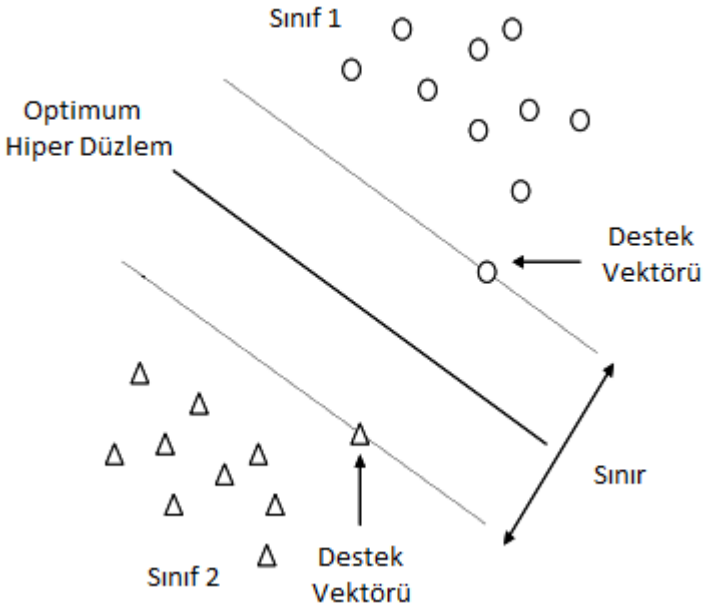
Şekil 3.3. Örnek Bayes ağı şeması(URL1)

3.3.3. Destek vektör makineleri

İstatistiksel öğrenme teorisine ve denetimli öğrenme algoritmasına sahip olan Destek Vektör Makineleri (DVM) başlangıçta iki sınıflı ve doğrusal olarak ayrılabilen verileri sınıflandırmak amacıyla geliştirilmiştir (Boser v.d., 1992). Bunun için n boyutlu girdileri, yüksek boyutlu bir özellik alanına eşleyip veriyi görselleştirir ve ardından özellik alanındaki iki sınıfı bir hiper düzlem tanımlaması ile ayırır (Moavenian v.d., 2010).

Algoritmaya ait sınıflandırıcının oluşturulması için iki sınıfa ait gözlemler, birbirine paralel iki çizgi yardımıyla bir hiperdüzlem oluşturularak doğrusal olarak ayrılır. Bu şekilde bir ayrım için sonsuz farklı hiper düzlem oluşturulabilir ancak DVM

algoritması, destek vektörleri üzerinden çizilen doğrular arasındaki uzaklığı maksimum yapacak hiperdüzlemi belirlemeye çalışır.



Şekil 3.4. Doğrusal bir şekilde ayrılabilen Destek Vektör Makineleri (Chandaka v.d., 2009).

İki farklı sınıfa ait gözlemler arasında diğer sınıfa en yakın gözleme destek vektörü denilmektedir ve her sınıf birden fazla destek vektörüne sahip olabilir. Farklı sınıflara ait destek vektörleri arasındaki mesafeye sınır denir ve bu mesafenin maksimum olduğu durumda sınırın tam ortasında bulunan, farklı sınıfların destek vektörlerine eşit mesafede olan, sınıfları birbirinden ayıran çizgiye ise optimal hiper düzlem denilmekte ve şekil 3.4.'te gösterilmektedir.

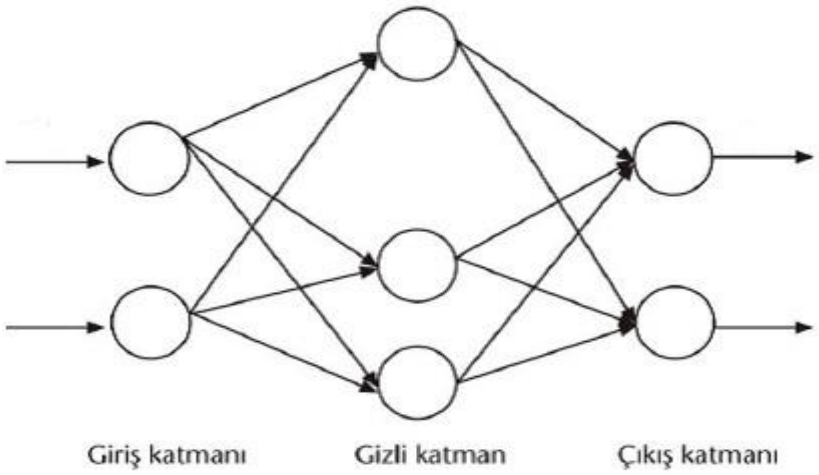
İlk olarak iki sınıf değerine sahip veri setleri için tasarlanan Destek Vektör Makineleri, zamanla doğrusal olmayan ve ikiden çok sınıf değerine sahip veri setlerinde çalışabilmesi için de tasarlanmıştır. Özellikle aykırı gözlemlere sahip veri setlerinde, sınıfları doğrusal bir şekilde ayırmak mümkün olmamaktadır ve bu şekilde doğrusal olarak ayrıştırılamayan veri setleri, algoritmanın çalışabilmesi için verilerin görselleştirildiği eğitim setinin ayrılabileceği daha üst bir özellik alanına işaretlenerek sınıflar birbirinden doğrusal olarak değil düzlemsel olarak ayırılır (Chen v.d., 2011).

Veri setinin aykırı gözlemlere sahip olması durumunda farklı sınıflara ait gözlemler diğer sınıfa ait gözlemlerle karışabilir ve iki sınıf heterojen olarak birbirinden ayrılamaz. Bu gibi bir durum Şekil 3.4.'te örneklendirilmiştir ve bu gibi durumlarda veri setindeki sınıfları ayırabilmek için gözlemler daha üst bir boyuta doğrusal olmayan haritalama ile dönüştürülerek, sınırı en büyük ayırıcı düzlem ile Destek Vektör Makineleri uygulanır.

Eğer bir veri seti ikiden fazla sınıf değerine sahipse bu sınıfları Destek Vektör Makineleri ile ayırabilmek için farklı yaklaşımlar geliştirilmiştir. Bu yaklaşımlar bire karşı bir yaklaşımı ve bire karşı hepsi yaklaşımıdır. Bire karşı hepsi yaklaşımı sürecinde sınıflardan birisi -1 ile, diğer tüm sınıf değerleri +1 ile yüklenerek destek vektörleri hesaplanır ve veri setindeki n adet sınıf için yine n adet sınıflandırma yapılırken bire karşı bir yaklaşımında ise her bir sınıf, diğer sınıflarla bire bir olarak test edilir ve maksimum kazanç stratejisi ile destek vektörleri seçilirken veri setindeki n adet sınıf için n*(n-1)/2 adet sınıflandırıcı kullanılmaktadır (Saigal v.d., 2019)

3.3.4. Yapay sinir ağları

Sinir ağları modeli insan biyolojisindeki sinir sisteminde, vücuda yapılan bir uyarının nöronlar yardımıyla beyne taşınması ve uyarının algılanıp ne olduğunun tanımlanması sürecinin matematiksel bir simulasyonudur. Ağ modelinde bilgiler nöron da denilen işleme elemanları arasında kurulan bağlantılar arasında aktarılır. Bu süreç bir giriş katmanından alınan verilerin, bir veya daha fazla gizli katmandan geçirilerek nihayetinde bir çıktı katmanına aktarılması ile sonuçlanır. Bilgilerin katmanlar arasında ilerleyiş süreci örneği şekil 3.5.'te gösterilmektedir.



Şekil 3.5. Örnek sinir ağı modeli

Yapay sinir ağlarının giriş katmanında bulunan nöronlar, veri setindeki özniteliklerin sayısı kadardır ve her nöron bir özniteliğe ait bilgileri taşımaktadır. Çıkış katmanında bulunan nöronlar ise veri setinin sınıf özniteliği içerisindeki sınıf değerlerini temsil etmektedir. Her katmanda bulunan nöron, kendisinden sonraki katmanda bulunan nöronla bağlantılı olabilir ancak kendi katmanı içerisindeki başka bir nöronla bağlantısı olamaz. Gizli katman içerisindeki nöronlar fonksiyon barındırmaktadır ve bu fonksiyon aşağıdaki şekilde formüle edilir:

$$\mathrm{NET} = \sum_{i,j}^{n} w_{ij} x_i + b \tag{3.14}$$

Denklem (3.14)'te belirtilen w_{ij}, bir bağlantının ağırlık değerini temsil etmektedir. x_i değeri ise bağımsız değişkenlerden birini temsil etmektedir. Algoritma, eğitim sürecinde kendisine verilen veri seti ile çalışarak her ağırlık değerini kendisinden sonra gelen katmanın hatalarına göre optimize eden bir geri yayılım algoritması şeklinde çalışır (Li v.d., 2017). Eğitim sürecinde edinilen bilgilerle çıkış katmanından geriye doğru daha önceki katmanlara geri dönerek ağırlıkların değiştirilmesi ile öğrenme ilerler ve en optimizasyon sağlanır. Artık algoritma test için kullanılan veri seti içerisindeki öznitelik değerlerine atanacak ağırlıklar ile işlem yapar ve çıkış katmanında nihai kararı olacak bir sınıf tahmin eder.

3.3.5. Lojistik regresyon

İstatistik biliminde doğrusal regresyon analizleri bir bağımlı değişken(Y) ile bir veya birden fazla bağımsız değişken(X) arasındaki sebep-sonuç ilişkisini belirlemek ve bu ilişki ile birlikte konu ile ilgili tahminler ve çıkarımlar yapabilmek amacıyla yapılmaktadır. Bağımlı ve bağımsız değişkenler, sürekli veya kategorik olabilmektedir. Analiz için kurulan ilk modelde bulunan bağımsız değişkenlerden, bağımlı değişken ile ilişkisi bulunmadığı tespit edilenler modelden çıkarılmaktadır. Bu sayede model daha öz ve açıklayıcı olmaktadır.

Lojistik regresyon, doğrusal regresyon ile benzerlik göstermesine karşın değişkenler nominal veya ikili olarak ele alındığında kullanılabilmektedir. Aksi durumlarla karşılaşıldığında model, bağımlı değişkenin sahip olduğu gerçel sayı değerlerini olasılıksal değerlere dönüştürerek işlem yapmak için kümülatif olasılık yoğunluk işlevini kullanır (Chen v.d., 2014). Sürekli verilerin kesikli verilere dönüştürülmesi süreci ile Bayes sınıflandırıcılar ile benzerlik göstermektedir.

3.3.6. k-en yakın komşuluk

Veri setindeki gözlemlerin çok boyutlu bir özellik uzayına yerleştirildiği bu algoritmada yeni bir gözlemin sınıf tahmini için ona en yakın nesnelerden faydalanılır. Gözlemler arasındaki mesafe genellikle Öklid formülü ile hesaplanır ve yeni gözlemin en yakın olduğu gözlemlerin sınıfları, yeni gözlemin sınıf tahmin değerini etkiler. Algoritmanın isminde bulunan k ibaresi, sınıflandırma sürecinde yeni gözleme en yakın olan kaç nesnenin algoritmaya dâhil olacağını belirler. Örneğin k=1 için yeni gözleme en yakın nesne sınıf değerini belirlerken, k=3 için yeni gözleme en yakın 3 nesne bulunur ve bu nesnelere ait sınıf değerleri incelenir. Yeni gözlem sisteme eklendiğinde ona en yakın olan nesneleri bulmak için algoritma her nesne ile olan mesafeyi hesaplar ve en düşük olanları seçer. Bu hesaplamadan sonra yeni gözleme en yakın olan nesneler içerisinde hangi sınıf değeri daha fazla ise o sınıf tahmin edilerek nihai atama yapılır. Burada k değerinin seçimi önem arz etmektedir, öyle ki k değeri eğer çift bir sayı olarak seçilirse sınıflara ait gözlem

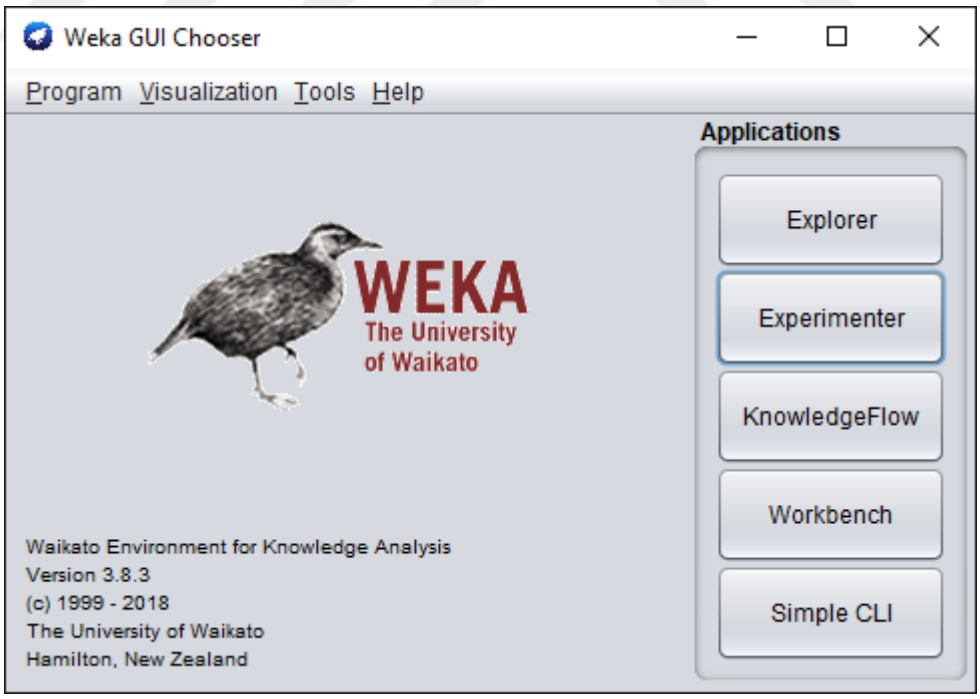
sayıları eşit olabilecektir. k değerinin tek sayı seçilmesi bu sorunu ortadan kaldıracaktır.

4. UYGULAMA

Tezin bu bölümü iki kısımdan oluşmaktadır. Birinci kısmında sınıflandırma yöntemlerinin performanslarını karşılaştırmak amacıyla, değişik sınıf ve değişken sayılarında 100 rasgele sınıflandırma verisi üretilmiştir. İkinci kısımda ise sınıflandırma yöntemlerinin performanslarını karşılaştırmak için çevrimiçi veri depoları olan UCI ve Kaggle'dan 100 adet gerçek veri seti elde edilmiştir.

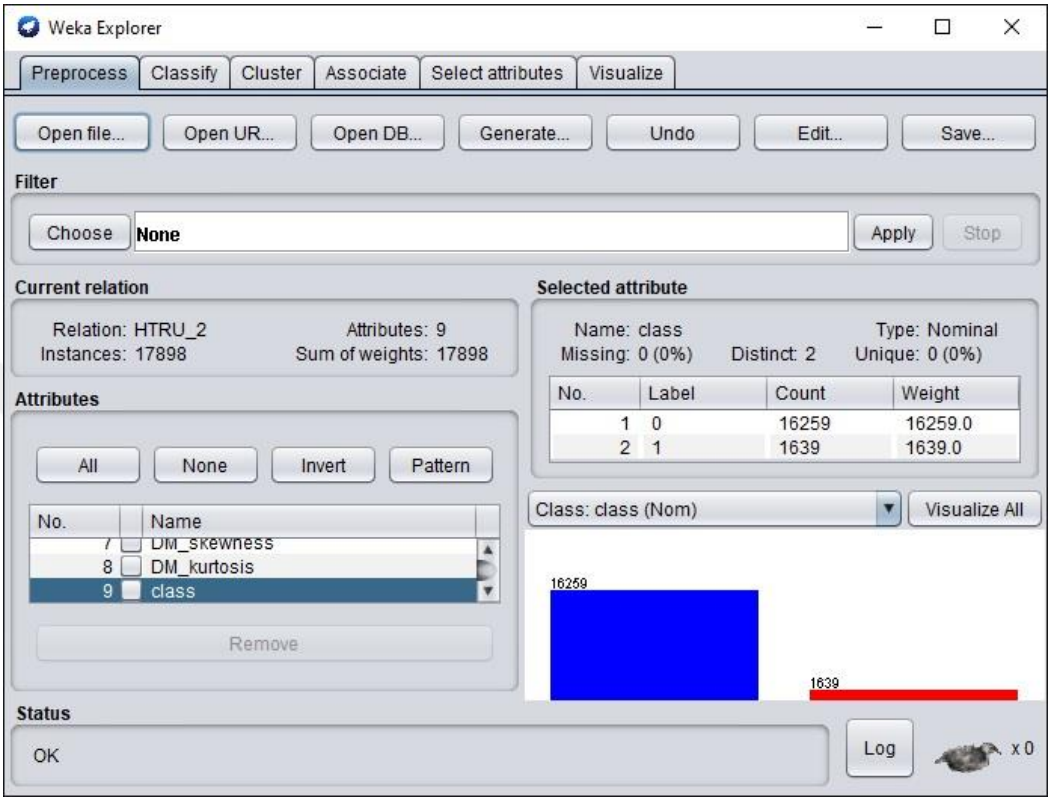
Veri setlerinin sınıflandırılması için açık kaynak kodlu bir sınıflandırma uygulaması olan WEKA'nın 3.8.3 versiyonu kullanılmıştır. Veri setleri, model oluşturulurken Hold-Out yöntemi ile %70 eğitim verisi ve %30 test verisi olarak ayrılmıştır. Uygulamada bulunan 41 adet sınıflandırma algoritması ile çalışılmış ve bu sınıflandırıcıların doğru sınıflandırılmış örnek oranları açısından başarıları karşılaştırılmıştır.

WEKA 3.8.3 yazılımında izlenen adımlar şu şekilde özetlenebilir.



Şekil 4.1. WEKA uygulaması açılış ekranı

WEKA uygulaması açıldığında, sınıflandırma algoritmalarına ulaşmak amacıyla "Explorer" menüsü kullanılmaktadır. Explorer menüsü açıldığında Şekil 4.2.'de verilen Preprocess ekranı ile karşılaşılmaktadır.

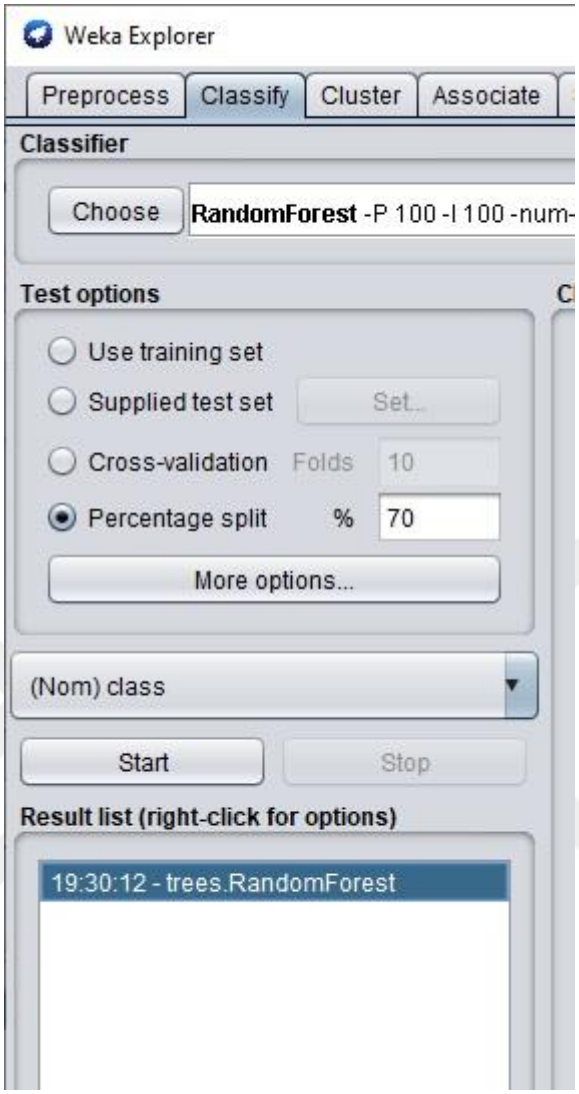


Şekil 4.2. WEKA Preprocess penceresi

WEKA Explorer arayüzü içerisinde farklı veri işleme ve veri madenciliği teknikleri için farklı sekmeler bulunmaktadır. İlk sekme olan "Preprocess" alanında veri madenciliği için kullanılacak veri setinin seçilmesi ve önişlemesi yapılır. Burada bulunan "Open File", "Open URL" ve "Open DB" ile farklı yerlerden veri seti tanıtma işlemi yapılabilmektedir. "Edit" butonu veri setine ait öznitelikleri, özniteliklerin veri türlerini ve gözlemleri görmemizi, sıralamamızı sağlar. "Choose" butonu veri önişleme için kullanılacak yöntemi ve hangi öznitelikler için geçerli olacağını seçmemize yarayan birimdir. "Current relation" bölmesi içerisinde veri setinin ismi, gözlem sayısı ve öznitelik sayısı belirtilmektedir. "Attributes" bölmesi içerisinde veri setinde bulunan öznitelikler bulunmaktadır ve istenmeyen öznitelikler bu alan içerisinde veri madenciliği sürecinden çıkarılabilmektedir. "Selected attribute" bölmesi sol taraftaki alandan o an seçili olan özniteliğin ismini, veri tipini,

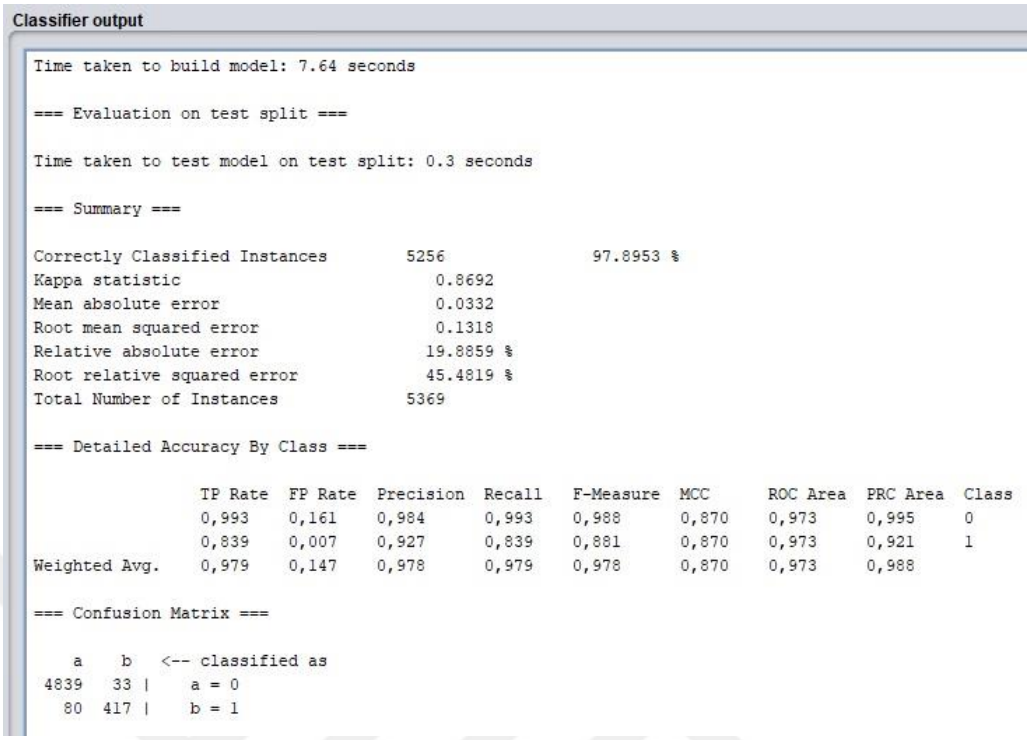
ne kadar ve ne oranda kayıp gözlem içerdiğini, içerisinde birbirinden farklı kaç adet gözlem içerdiğini gösterir. Yine aynı bölüm içerisinde seçilmiş öznitelik eğer numerik ise bu veri setine ait minimum, maksimum, ortalama ve standart sapma değerleri görülebilirken, seçilmiş öznitelik eğer kategorik ise bu öznitelik içerisindeki kategorik değerler ve bu değerlerin kaç adet olarak bulundukları bilgisi görülebilir. Pencerenin sağ alt kısmında ise seçilen özniteliğin görselleştirilmiş hâli incelenebilmektedir.

Uygulama içerisindeki ikinci sekme olan "Classify" sınıflandırma işlemlerinin yapıldığı kısımdır. İlk sekmede seçilen veri seti üzerinde uygulanacak sınıflandırma işlemleri bu sekme içerisinde yapılmaktadır. Bu sekmede Classifier içerisinde bulunan "Choose" butonu, uygulayacağımız sınıflandırıcıyı seçmemizi sağlar. Bu alanda 7 ana başlık altında 56 adet sınıflandırıcı bulunmaktadır. "Test options" alanı içerisinde, model kurulurken uygulanacak yöntem seçilmektedir. Cross-validation ile çapraz doğrulama yapılabilmektedir. Burada bulunan folds ibaresi, çapraz doğrulama işlemindeki k ifadesini temsil etmektedir. 10 olarak seçilen folds ifadesi ile işlem yapmak, 10 kat çapraz doğrulama ile model oluşturmak anlamına gelmektedir. Percentage split kısmında ise Hold-Out ile model oluşturulabilmektedir. Burada girilebilen değer, veri setinin ne kadarlık bir oranının eğitim verisi olarak kullanılacağını belirtmektedir. Bu alanın hemen altında ise veri setine ait öznitelikler içerisinden sınıf değişkenlerinin bulunduğu öznitelik seçilerek sınıflandırmanın bu öznitelik üzerinden yapılması sağlanır. "Start" butonu ile sınıflandırma işlemi başlar ve öncelikle model oluşturulur, ardından bu model test verilerinin sınıf değerlerinin tahmini için uygulanır. Şekil 4.3., Weka yazılımının "classify" bölmesine ilişkin ana ekranı göstermektedir.



Şekil 4.3. WEKA Classify penceresi ayarlar bölmesi

Yine "Classify" sekmesi içerisinde bulunan "Classifier output" penceresi içerisinde sınıflandırma sürecine ait bilgiler bulunmaktadır. Bu pencereye ait bir görüntü Şekil 4.4.'te verilmektedir.



Şekil 4.4. Sınıflandırma algoritmasına ait çıktı ekranı

Şekil 4.4. incelendiğinde, seçili algoritma için eğitim verisi ile sınıflandırma algoritması modelinin oluşturulması 7.64 saniye, bu modelin test verilerine uygulanması süreci ise 0.3 saniye sürmüştür. Summary kısmında ise ilk olarak doğru sınıflandırılmış örneklere ilişkin sayı ve oran bilgileri yeralmaktadır. Kullanılan veri seti ve algoritma için doğru sınıflandırılmış örnek sayısı 5256, doğru sınıflandırılmış örnek oranı ise %97.8953 olarak belirtilmiştir.

Yine bu bölümde, sınıflandırma algoritmasının başarısını karşılaştırmada kullanılabilecek Kappa istatistiği, ortalama mutlak hata, karekök ortalama mutlak hata kareler, göreceli mutlak hata, karekök göreceli hata kareler ve toplam gözlem sayısı gösterilmektedir. Detailed Accuracy By Class bölümünde ise sınıflara ait gerçek pozitif, yanlış pozitif, hassasiyet, geri çağırma, F-değeri gibi sınıflandırma başarısı için kullanılabilecek diğer bilgiler verilmektedir.

En altta Confusion Matrix olarak ayrılmış bölümde ise veri setine ait sınıflar için karışıklık matrisi verilmiştir. Bu matris incelendiğinde, veri seti içerisinde sınıf değeri gerçekten a iken, sınıflandırıcı tarafından a olarak tahmin edilen değerlerin sayısının 4839 olarak görmekteyiz. Gerçek sınıf değeri a iken b olarak tahmin edilen

80, gerçek sınıf değeri b iken a olarak tahmin edilen 33 gözlem ve gerçek sınıf değeri b iken b olarak tahmin edilen 417 gözlem olduğunu görmekteyiz.

Sonraki bölüm simulasyon çalışmasının sonuçlarını içermektedir.

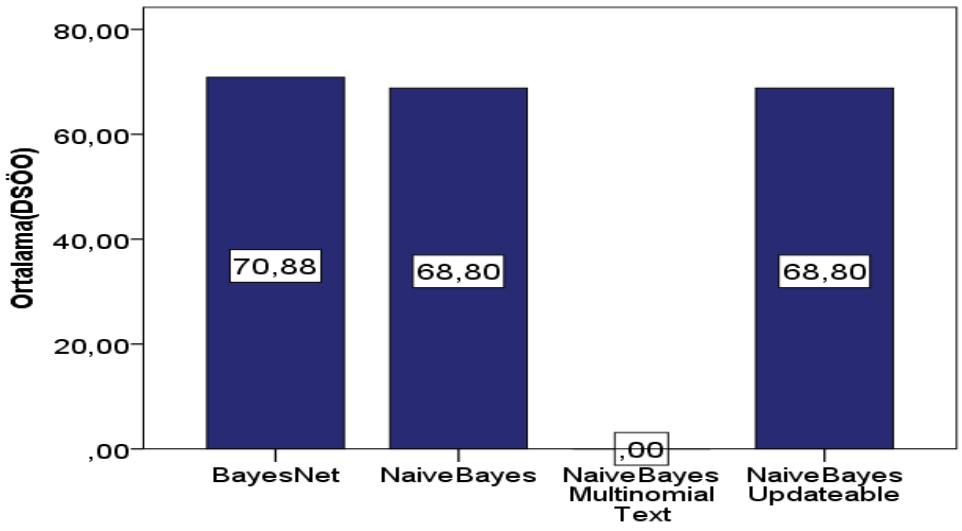
4.1 Simulasyon Çalışması

Bu bölümde WEKA 3.8.3 veri madenciliği yazılımında yer alan Bayes Net, RandomRBF ve RDG1 datagenerators yöntemleri kullanılarak üretilen 100 sınıflandırma veri setinden elde edilen sonuçlara yer verilmektedir. EK-1'de üretilen veri setlerinin özellikleri verilmektedir.

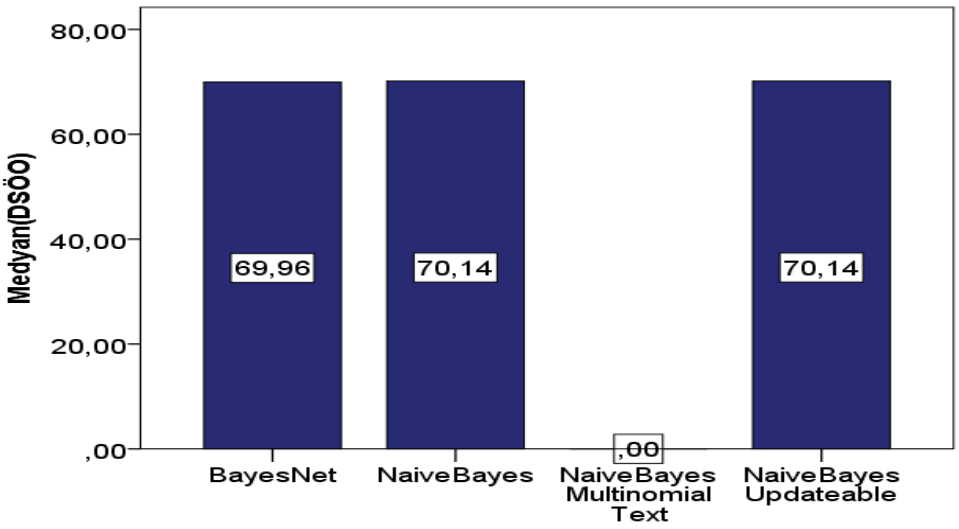
EK-1'de verilen veri setlerine Bayes kategorisinden 4, Function kategorisinden 4, Lazy kategorisinden 3, Meta kategorisinden 17, Misc kategorisinden 1, Rules kategorisinden 5, Trees kategorisinden 7 olmak üzere toplam 41 sınıflandırma yöntemi uygulanmıştır. Sınıflandırma yöntemlerinin kategorilerine göre sınıflandırma başarıları sonraki bölümlerde verilmektedir.

4.1.1 Simule edilen veriler için Bayes kategorisi sınıflandırma sonuçları

Bu bölümde simule edilen 100 veri setine Bayes net, Naive Bayes, Naive Bayes Multinomial Text, Naive Bayes Updateable sınıflandırıcıların uygulanması sonucunda elde edilen sonuçlara yer verilmektedir. Şekil 4.5. ve Şekil 4.6.'da bu sınıflandırıcılardan elde edilen DSÖO'ların sırasıyla ortalama ve medyan değerleri verilmektedir.



Şekil 4.5. Simulasyon verileri için Bayes sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.6. Simulasyon verileri için Bayes sınıflandırıcılarına ait DSÖO medyan değerleri

Bayes sınıflandırıcılara ait DSÖO ortalama ve medyan değerlerine bakıldığında, Naive Bayes Multinomial Text sınıflandırıcısının başarı oranının 0 olarak elde edildiği görülmektedir. Bunun dışında şekillerden, Naive Bayes ve Naive Bayes Updateable sınıflandırıcılarının aynı performansı sergilediği, en yüksek performansın ortalama değerine göre Bayes net sınıflandırıcısından, medyan değerine göre ise Naive Bayes ve Naive Bayes Updateable sınıflandırıcısından elde edildiği görülmektedir. Bayes tabanlı sınıflandırma yöntemlerinin sınıflandırma başarılarının istatistiksel açıdan anlamlı olup olmadığını test etmek amacıyla Wilcoxon testi gerçekleştirilmiştir. Çizelge 4.1.'de Wilcoxon testinin sonuçları verilmektedir.

Çizelge 4.1. Simule edilen veriler için Bayes sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

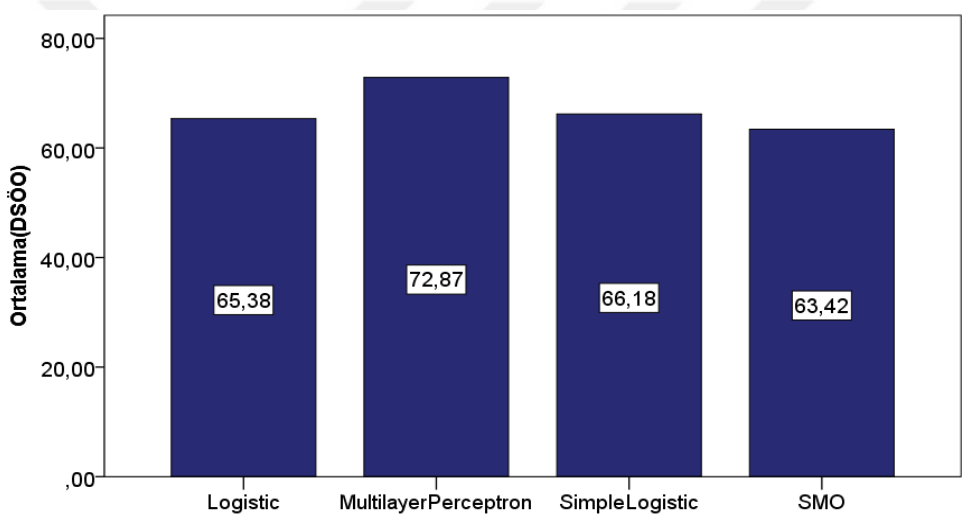
Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
Naive Bayes-Bayes Net	63	32	0.000
Naive Bayes Multinomial Text-Bayes Net	100	0	0.000
Naive Bayes Updateable-Bayes Net	63	32	0.000
Naive Bayes Multinomial Text-Naive Bayes	100	0	0.000
Naive Bayes Updateable-Naive Bayes	0	0	0.000
Naive Bayes Updateable-Naive Bayes Multinomial Text	0	100	0.000

Çizelge 4.1.'de p değerlerine bakıldığında tüm değerlerinin 0.05'ten küçük olduğu görülmektedir. Buradan Bayes kategorisindeki tüm sınıflandırma çiftlerinin performanslarının arasındaki farkın istatistiksel açıdan anlamlı olduğunu söylemek mümkündür. Çizelge 4.1.'de negatif rank sütunu sınıflandırma çifti sütununda ikinci sınıflandırma yönteminin performansının yüksek olduğu veri sayısını, pozitif rank

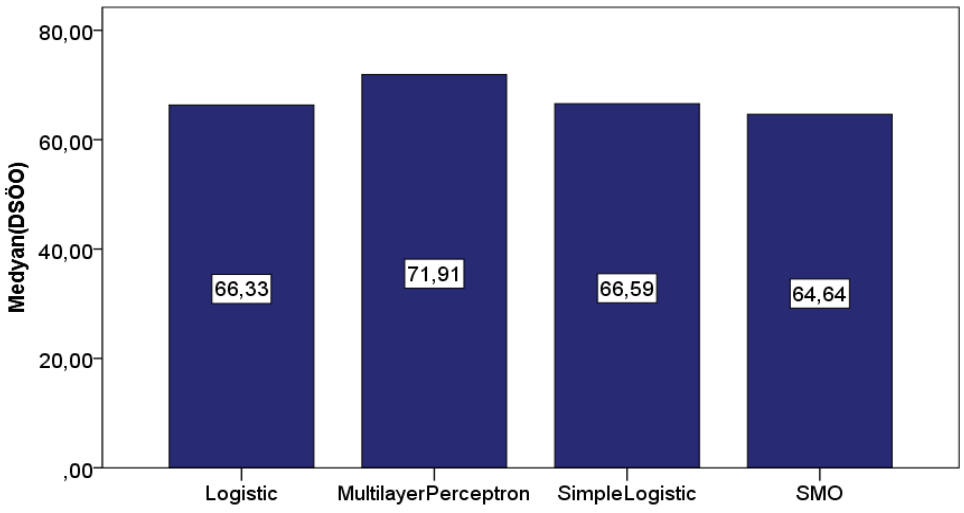
sütunu ise birinci sınıflandırma yönteminin başarılı olduğu veri sayısını göstermektedir. Buna göre 100 veri setinin 63'ünde Bayes net, 32'sinde ise Naive Bayes sınıflandırma yönteminin daha yüksek DSÖO değerine sahip olduğunu söylemek mümkündür. Diğer durumlar da benzer şekilde yorumlanabilir.

4.1.2 Simule edilmiş veriler için Function kategorisi sınıflandırma sonuçları

Bu bölümde, Logistic, Multilayer Perceptron, Simple Logistic ve SMO sınıflandırıcılarından elde edilen sonuçlar verilmektedir. Şekil 4.7. ve Şekil 4.8., 100 veri seti için bu 4 sınıflandırıcıdan elde edilen DSÖO'ların ortalama ve medyan değerlerini göstermektedir.



Şekil 4.7. Simulasyon verileri için Function sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.8. Simulasyon verileri için Function sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.7. ve Şekil 4.8. incelendiğinde, DSÖO değerlerinin hem ortalama hem de medyan değerlerine göre en düşük sınıflandırma başarısına sahip sınıflandırma yönteminin SMO, en yüksek başarıya sahip sınıflandırma yönteminin ise Multilayer Perceptron olduğu görülmektedir. Çizelge 4.2.'de Wilcoxon testine göre aralarında istatistiksel açıdan anlamlı farklılık bulunan sınıflandırma çiftleri verilmektedir.

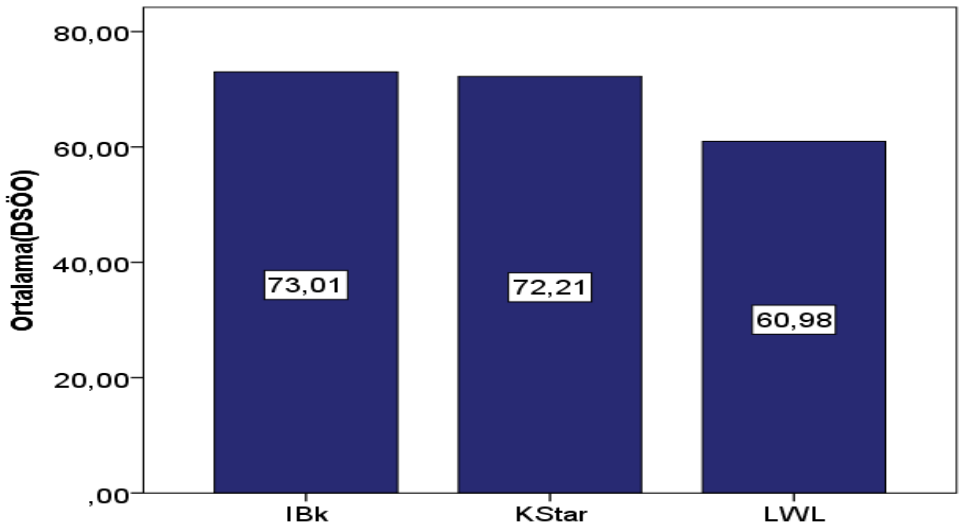
Çizelge 4.2. Simule edilen veriler için Function sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
Multilayer Perceptron-Logistic	14	83	0.000
SMO-Logistic	55	34	0.002
Simple Logistic- Multilayer Perceptron	78	18	0.000
SMO-Multilayer Perceptron	81	14	0.000
SMO-Simple Logistic	57	29	0.000

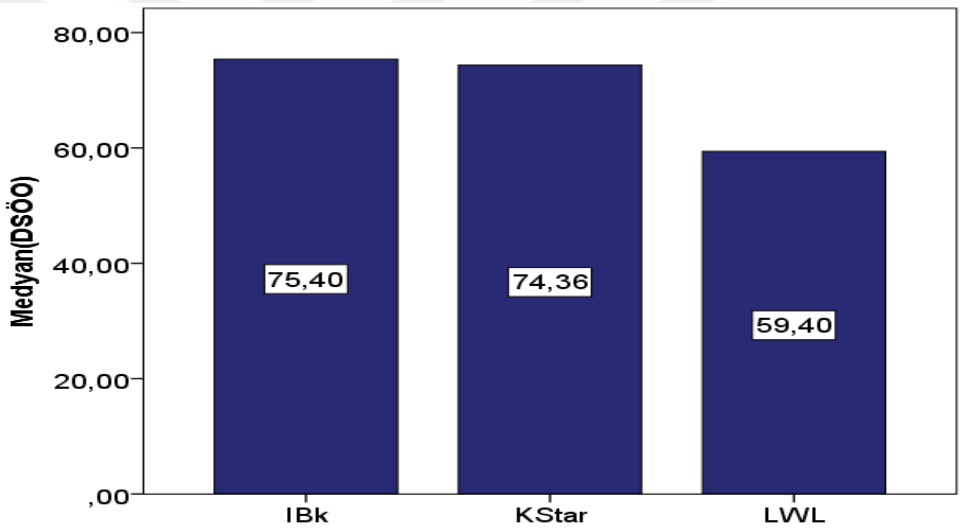
Çizelge 4.2.'ye göre Multilayer Perceptron-Logistic, SMO-Logistic, Simple Logistic-Multilayer Perceptron, SMO-Multilayer Perceptron ve SMO-Simple Logistic sınıflandırma çiftlerinin performanslarının arasındaki fark istatistiksel açıdan anlamlıdır. Pozitif rank sütunu sırasıyla ilk sınıflandırma yöntemlerinin negatif rank sütunu ise ikinci sınıflandırma yöntemlerinin başarılı olduğu veri seti sayısını göstermektedir. Buna göre Logistic ile karşılaştırıldığında Multilayer Perceptron sınıflandırma yöntemi 100 veri setinin 83'ünde daha yüksek DSÖO değeri sağlamıştır. Diğer sınıflandırma çiftleri için de benzer yorumlar yapılabilir.

4.1.3 Simule edilmiş veriler için Lazy kategorisi sınıflandırma sonuçları

Weka veri madenciliği yazılımı Lazy kategorisine ait IBk, KStar ve LWL sınıflandırıcılarına ilişkin ortalama ve medyan DSÖO değerleri sırasıyla Şekil 4.9. ve Şekil 4.10.'da verilmektedir.



Şekil 4.9. Simulasyon verileri için Lazy sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.10. Simulasyon verileri için Lazy sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.9. ve Şekil 4.10.'a bakıldığında, Lazy kategorisindeki sınıflandırıcılar arasından sınıflandırma başarısı en yüksek olan yöntemin IBk, en düşük olan yöntemin ise LWL olduğu görülmektedir. Bu kategorideki sınıflandırma yöntemlerine ilişkin Wilcoxon test sonuçları ise Çizelge 4.3.'te verilmektedir.

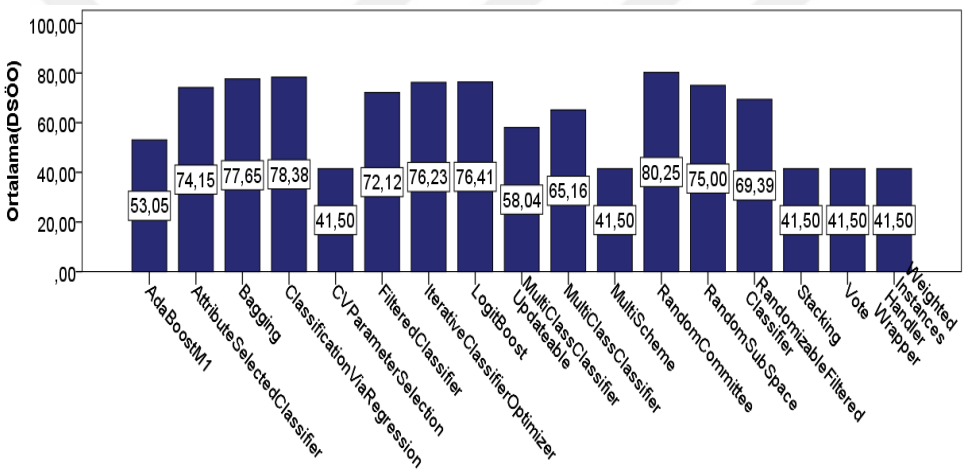
Çizelge 4.3. Simule edilen veriler için Lazy sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
LWL-IBk	68	29	0.000
LWL-KStar	68	27	0.000

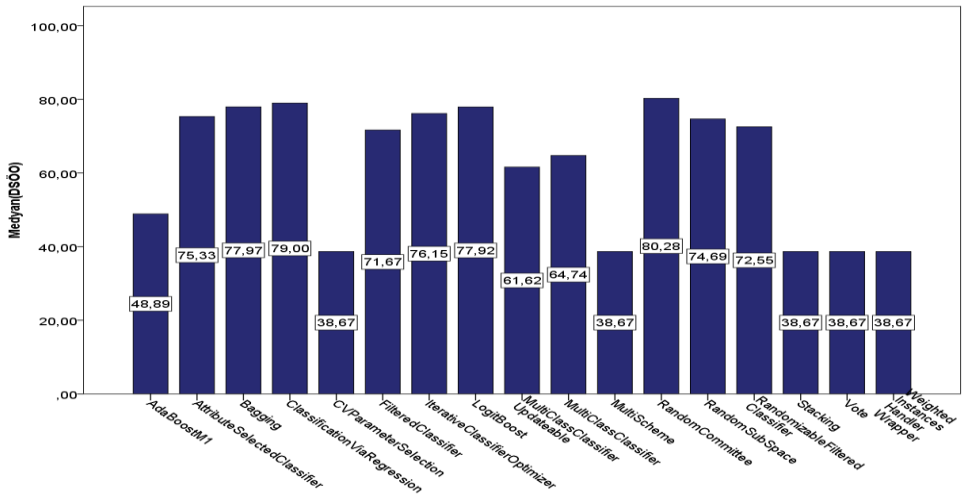
Çizelge 4.3.'e göre, LWL-IBk sınıflandırma yöntemlerinin karşılaştırılmasında, 100 veri setinin 68'inde IBk yöntemi, 29'uda LWL yöntemi daha başarılı sonuçlar vermiştir. LWL-KStar sınıflandırma yöntemlerinin karşılaştırılmasında ise, 100 veri setinin 68'inde KStar yöntemi, 27'sinde LWL yöntemi daha yüksek DSÖO değeri sağlamıştır.

4.1.4 Simule edilmiş veriler için Meta kategorisi sınıflandırma sonuçları

Şekil 4.11. ve Şekil 4.12.'de Meta kategorisinde yer alan 17 sınıflandırma yönteminin simule edilen 100 veri setine uygulanması sonucunda elde edilen DSÖO değerlerinin ortalama ve medyan değerlerini göstermektedir.



Şekil 4.11. Simulasyon verileri için Meta sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.12. Simulasyon verileri için Meta sınıflandırıcılarına ait DSÖO medyan değerleri

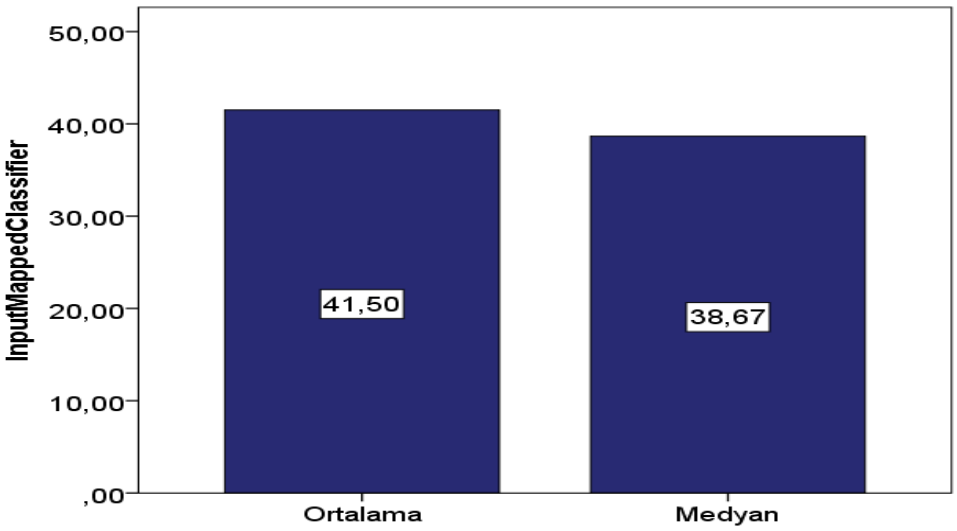
Şekil 4.11. ve Şekil 4.12. incelendiğinde, Random Commitee sınıflandırıcısının en başarılı performansı sergilediği, CV Parameter Selection, Multi Scheme, Stacking, Vote ve Weighted Instances Handler Wrapper sınıflandırıcılarının ise en düşük sınıflandırma başarısına sahip olduğu görülmektedir.

Meta sınıflandırıcılara ilişkin test sonuçları EK-2'de verilmektedir. EK-2'de elde edilen bazı sonuçlar şu şekilde özetlenebilir.

- Random Committee sınıflandırıcısı ile diğer tüm sınıflandırıcıların performansı arsındaki fark istatistiksel açıdan anlamlıdır ve Random Committee'nın sınıflandırma başarısı daha yüksektir.
- Wilcoxon testine göre, Classification Via Regression sınıflandırıcısı, Bagging sınıflandırıcısı ile benzer, Random Committee dışındaki tüm sınıflandırıcılardan daha yüksek sınıflandırma performansı sağlamıştır.
- CV Parameter Selection, Multi Scheme, Stacking, Vote ve Weighted Instances Handler Wrapper sınıflandırıcısı tüm veri setlerinde aynı DSÖO değerlerini vermiştir.

4.1.5 Simule edilmiş veriler için Misc kategorisi sınıflandırma sonuçları

Weka veri madenciliği yazılımında Misc kategorisinden yalnızca Input Mapped Classifier sınıflandırıcısı seçilmiştir. Şekil 4.13. Input Mapped Classifier sınıflandırıcısından elde edilen DSÖO değerlerinin ortalama ve medyan değerini göstermektedir.

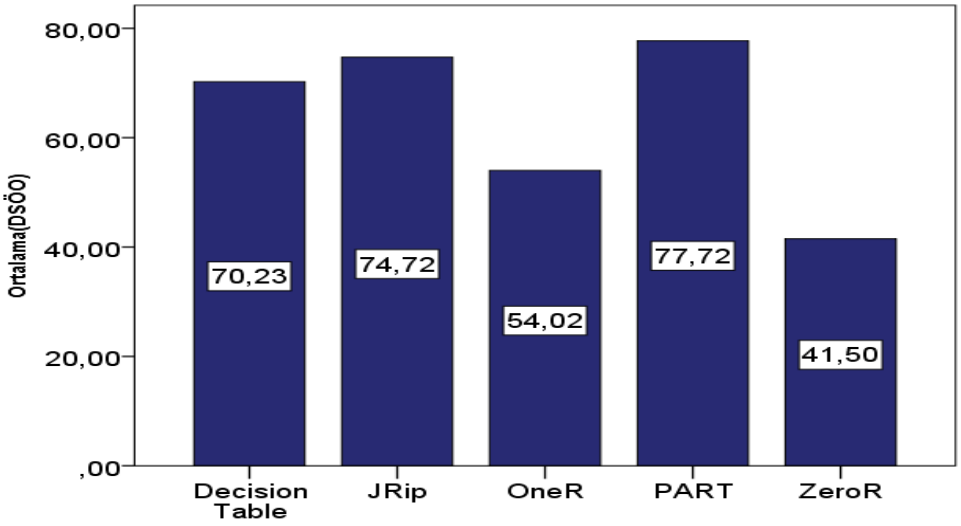


Şekil 4.13. Simulasyon verileri için Misc sınıflandırıcılarına ait DSÖO ortalama ve medyan değerleri

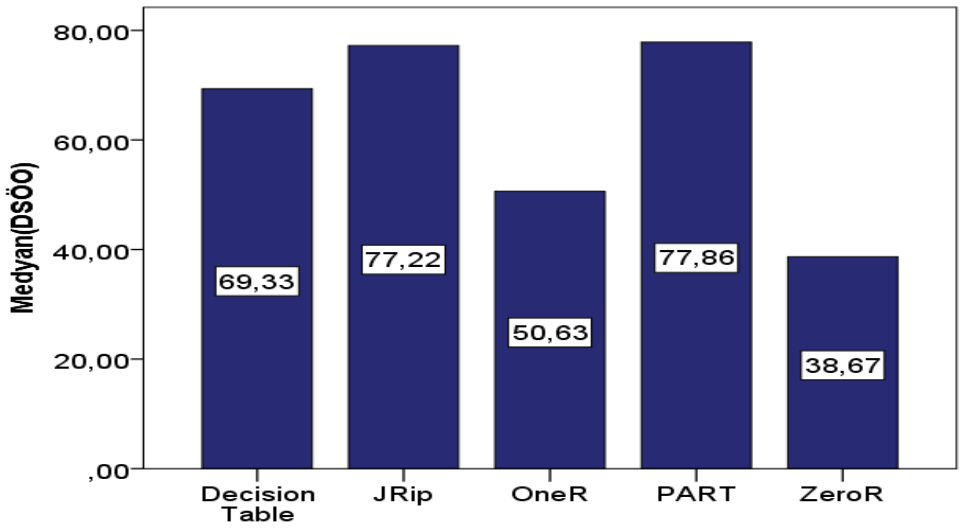
Input Mapped Classifier sınıflandırıcısı ortalamada %41.5, medyanda ise %38.67 başarı sağlamıştır.

4.1.6 Simule edilmiş veriler için Rules kategorisi sınıflandırma sonuçları

Rules kategorisinden 5 sınıflandırıcı seçilmiştir. Bu sınıflandırıcılara ilişkin ortalama ve medyan DSÖO değerleri sırasıyla Şekil 4.14. ve Şekil 4.15.'te verilmiştir.



Şekil 4.14. Simulasyon verileri için Rules sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.15. Simulasyon verileri için Rules sınıflandırıcılarına ait DSÖO medyan değerleri

Rules sınıflandırıcılara ait Şekil 4.14. ve Şekil 4.15. incelendiğinde PART sınıflandırıcısının %77.7 ortalama ve %77.86 medyan değeri ile en başarılı, ZeroR sınıflandırıcısının ise %41.5 ortalama, %38.65 medyan değeri ile en başarısız sınıflandırıcı olduğu görülmektedir. Çizelge 4.4. Wilcoxon test sonuçlarını vermektedir.

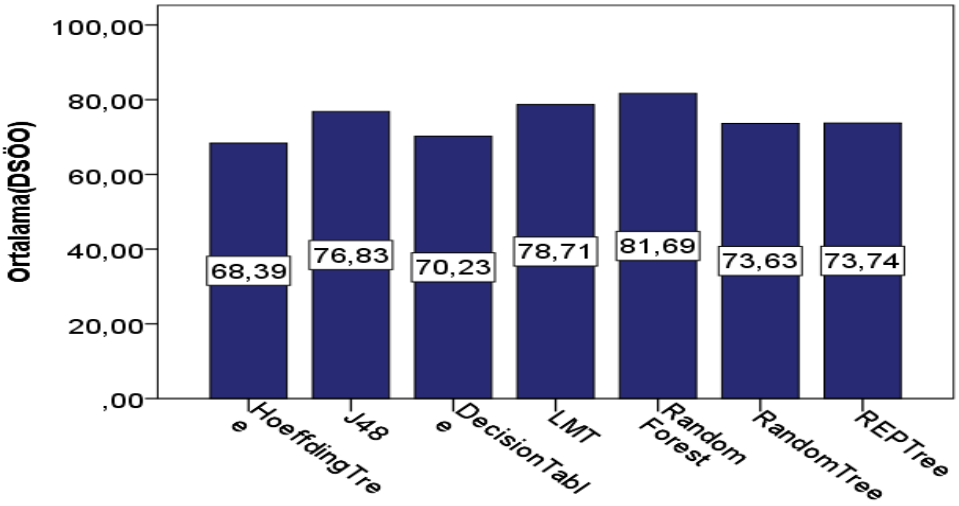
Çizelge 4.4. Simule edilen veriler için Rules sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
JRip-DecisionTable	24	67	0.000
OneR-DecisionTable	88	8	0.000
PART-DecisionTable	10	83	0.000
ZeroR-DecisionTable	91	1	0.000
OneR-JRip	94	2	0.000
PART-JRip	25	69	0.000
ZeroR-JRip	98	1	0.000
PART-OneR	5	94	0.000
ZeroR-OneR	89	10	0.000
ZeroR-PART	97	1	0.000

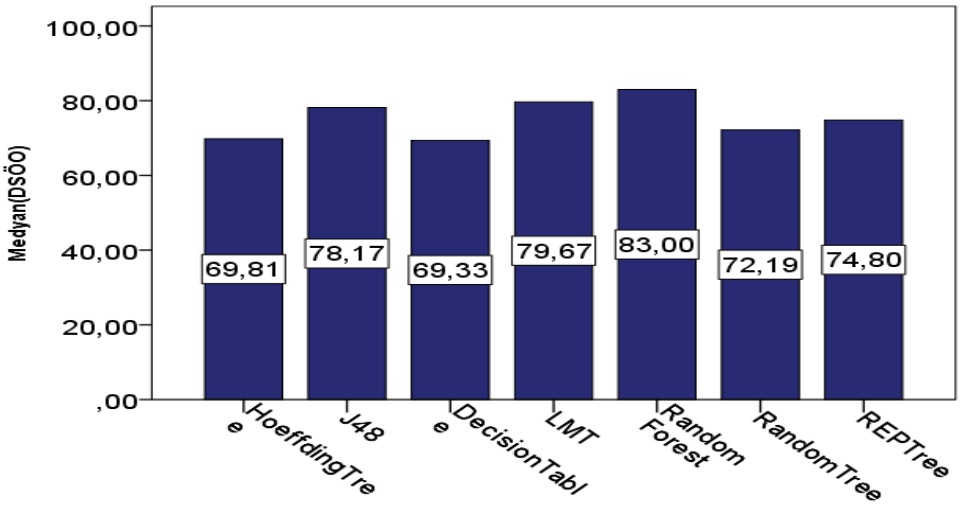
Çizelge 4.4.'e göre, tüm sınıflandırma çiftlerinin performansları arasındaki fark istatistiksel açıdan anlamlı bulunmuştur. Decision Table, OneR ve ZeroR sınıflandırıcıları ile kıyaslandığında daha fazla veri seti için daha yüksek bir başarı sağlamıştır. JRip, PART dışındaki tüm sınıflandırıcılardan daha yüksek bir performans sergilemiştir. OneR sınıflandırıcısı, ZeroR sınıflandırıcısı ile karşılaştırıldığında 100 veri setinden 89'unda daha başarılı sonuçlar vermiştir.

4.1.7 Simule edilmiş veriler için Trees kategorisi sınıflandırma sonuçları

Bu bölümde, rasgele üretilen 100 veri seti kullanılarak, Trees kategorisinden 7 sınıflandırıcının performansı karşılaştırılmıştır. Şekil 4.16. ve Şekil 4.17. sırasıyla Trees kategorisindeki sınıflandırıcılardan elde edilen ortalama ve medyan DSÖO değerlerini göstermektedir.



Şekil 4.16. Simulasyon verileri için Trees sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.17. Simulasyon verileri için Trees sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.16. ve Şekil 4.17.'ye bakıldığında, Random Forest sınıflandırıcısının en yüksek DSÖO değerlerine, Hoeffding Tree sınıflandırıcısının ise en düşük DSÖO değerlerine sahip olduğu görülebilir. Çizelge 4.5.'te Trees kategorisindeki sınıflandırıcılar için Wilcoxon test sonuçları verilmektedir.

Çizelge 4.5. Simule edilen veriler için Trees sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
J48-Hoeffding Tree	16	79	0.000
DecisionStump-HoeffdingTree	33	60	0.038
LMT-HoeffdingTree	4	90	0.000
RandomForest-HoeffdingTree	5	93	0.000
RandomTree-HoeffdingTree	25	72	0.000
REPTree-HoeffdingTree	25	72	0.000
DecisionStump-J48	79	10	0.000
LMT-J48	29	55	0.000
RandomForest-J48	15	78	0.000
RandomTree-J48	62	32	0.001
REPTree-J48	58	34	0.000
LMT-DecisionStump	12	86	0.000
RandomForest-DecisionStump	11	88	0.000
RandomTree-DecisionStump	26	68	0.000
REPTree-DecisionStump	20	73	0.000
RandomForest-LMT	22	70	0.000
RandomTree-LMT	72	20	0.000
REPTree-LMT	69	24	0.000
RandomTree-RandomForest	87	5	0.000
REPTree-RandomForest	83	12	0.000
REPTree-RandomTree	44	52	0.623

Çizelge 4.5.'e bakıldığında, p değerleri 0.05'ten küçük olduğu için REPTree-RandomTree sınıflandırma çifti dışındaki tüm sınıflandırma çiftlerinin performansları arasındaki farkın istatistiksel açıdan anlamlı olduğu görülmektedir.

4.1.8 Simule edilmiş veriler için genel karşılaştırma sonuçları

Bu bölümde, simule edilmiş veri setleri için her bir sınıflandırma kategorisindeki en iyi performansa sahip sınıflandırma yöntemlerinin performansları karşılaştırılmıştır. Çizelge 4.6.'ya en iyi performansa sahip sınıflandırma yöntemlerine ilişkin Wilcoxon test sonuçları verilmektedir.

Çizelge 4.6. Simule edilen veriler ve en iyi sınıflandırıcılar için Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
MultiLayerPerceptron-BayesNet	42	52	0.080
IBk-BayesNet	45	53	0.293
RandomCommittee-BayesNet	10	86	0.000
PART-BayesNet	11	79	0.000
RandomForest-BayesNet	9	88	0.000
IBk-MultilayerPerceptron	50	48	0.701
RandomCommittee-MultiLayerPerceptron	19	78	0.000
PART-MultiLayerPerceptron	34	63	0.000
RandomForest-MultiLayerPerceptron	7	86	0.000
RandomCommittee-IBk	43	55	0.000
PART-IBk	47	51	0.015
RandomForest-IBk	34	59	0.000

Çizelge 4.6.(Devam)

PART-RandomCommittee	67	26	0
RandomForest-RandomCommittee	26	59	0
RandomForest-PART	21	71	0

Çizelge 4.6.'ya bakıldığında, MultiLayer Perceptron-Bayes Net, IBk-Bayes Net, IBk-MultiLayer Perceptron dışındaki tüm sınıflandırma çiftlerinin performansları arasındaki farkın anlamlı olduğu görülebilir. En yüksek ortalama ve medyan DSÖO değerlerini veren Random Forest-Random Committee sınıflandırma çifti incelendiğinde, 100 veri setinin 59'unda RandomForest, 26'sında RandomCommitte'nin daha başarılı olduğu, 15'inde ise her iki yöntemden de aynı DSÖÖ değerleri elde edildiği görülebilir. Burada daha fazla veri seti için iyi sonuç veren Random Forest yöntemi olduğu ve Random Forest-Random Committee sınıflandırma çiftinin performansları arasındaki fark istatistiksel açıdan anlamlı olduğu için simule edilmiş veri setlerinde en başarılı sınıflandırıcının Random Forest olduğunu söylemek mümkündür.

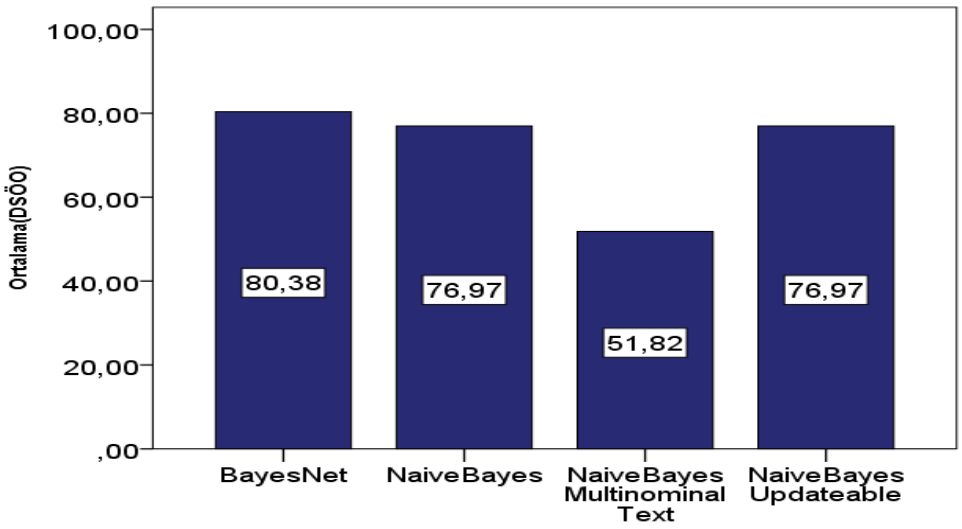
4.2 Sınıflandırma yöntemlerinin gerçek veri setlerindeki performanslarının karşılaştırılması

Çalışmada kullanılan gerçek veri setlerine ait özellikler EK-3'te verilmiştir. Bu çizelgeye göre elde edilen verilere ait gözlem sayıları 15 ile 32561 arasında, öznitelik sayıları 4 ile 206 arasında, sınıf sayıları ise 2 ile 30 arasında farklılık göstermektedir.

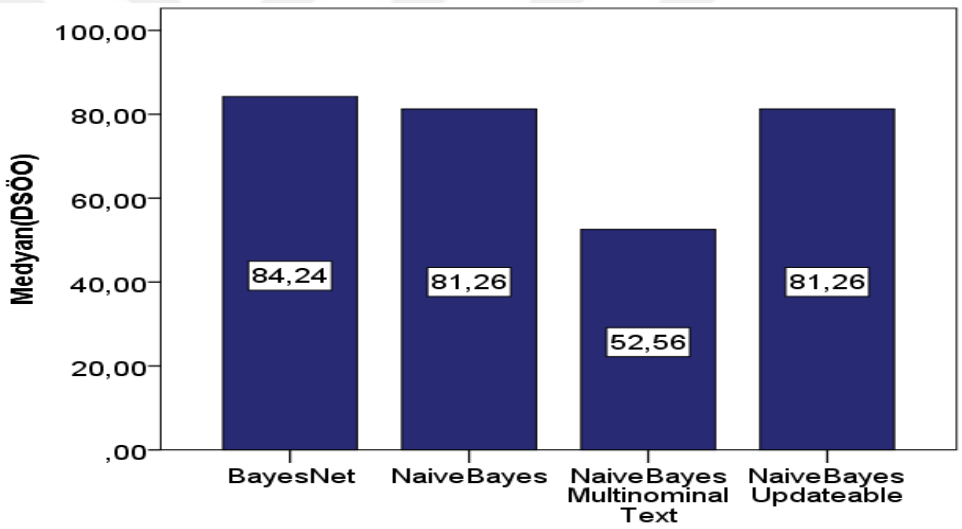
Sonraki bölümde, gerçek veri setleri için her bir kategorideki sınıflandırma yöntemlerinin ayrı ayrı performansları incelenmiştir.

4.2.1 Gerçek veri setleri için Bayes kategorisi sınıflandırma sonuçları

Şekil 4.18. ve Şekil 4.19.'da Bayes kategorisindeki sınıflandırıcılardan gerçek veri setleri için elde edilen ortalama ve medyan DSÖO değerleri verilmektedir.



Şekil 4.18. Gerçek veriler için Bayes sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.19. Gerçek veriler için Bayes sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.18. ve Şekil 4.19.'a bakıldığında, gerçek veri setlerine Bayes kategorisindeki sınıflandırma yöntemlerinin uygulanması sonucunda, en başarılı sınıflandırma yönteminin Bayes Net, en kötü performansa sahip sınıflandırma yönteminin ise Naive Bayes Multinominal Text olduğu görülebilir. Bunun dışında, Naive Bayes ile Naive Bayes Updateable sınıflandırma yönteminin aynı performansı sergilediği tespit edilmiştir. Bayes kategorisindeki sınıflandırıcıların performansları arasında istatistiksel açıdan anlamlı bir farklılık olup olmadığını tespit etmek amacıyla simulasyon verileri ile benzer bir şekilde Wilcoxon testi gerçekleştirilmiştir. Test sonuçları Çizelge 4.7.'de verilmektedir.

Çizelge 4.7. Gerçek veri setleri için Bayes sınıflandırıcılara ilişkin Wilcoxon test sonuçları

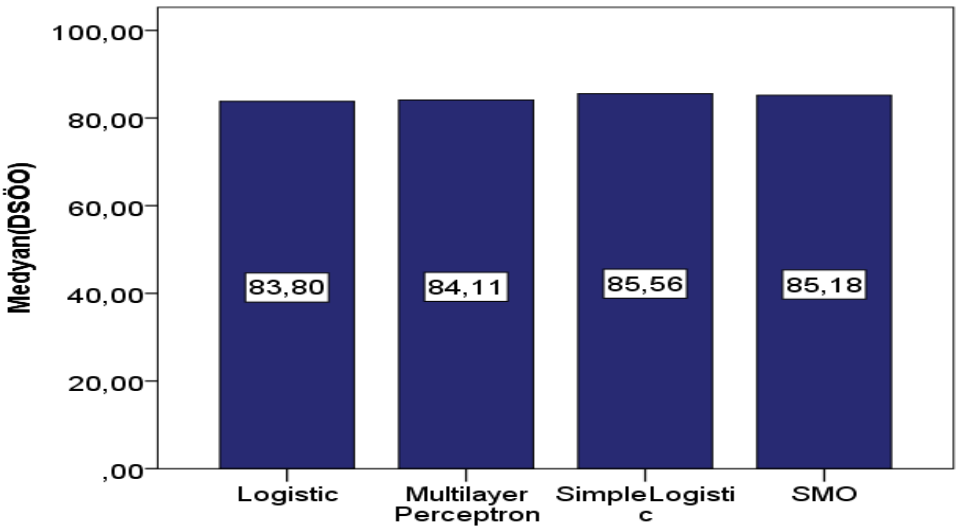
Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
NaiveBayes-BayesNet	54	27	0.000
NaiveBayesMultinominalText-BayesNet	89	7	0.000
NaiveBayesUpdateable-BayesNet	54	27	0.000
NaiveBayesMultinominalText-NaiveBayes	83	13	0.000
NaiveBayesUpdateable-NaiveBayes	0	0	1.000
NaiveBayesUpdateable-NaiveBayesMultinominalText	13	83	0.000

Çizelge 4.7.'ye göre NaiveBayesUpdateable-NaiveBayes sınıflandırma çifti dışındaki tüm sınıflandırma çiftlerinin performansları arasında anlamlı bir farklılık vardır. Buna göre, BayesNet'in sınıflandırma başarısının NaiveBayes, NaiveBayesMultinominalText, NaiveBayesUpdateable'dan, NaiveBayes ve NaiveBayesUpdateable sınıflandırma yöntemlerinin sınıflandırma başarılarının ise NaiveBayesMultinominalText'den anlamlı bir şekilde yüksek olduğu söylenebilir.

4.2.2 Gerçek veri setleri için Function kategorisi sınıflandırma sonuçları

Şekil 4.20. ve Şekil 4.21. sırasıyla gerçek veri setleri için Function kategorisindeki sınıflandırıcılardan elde edilen ortalama ve medyan DSÖO değerlerini göstermektedir.

Şekil 4.20. Gerçek veriler için Function sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.21. Gerçek veriler için Function sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.20. ve Şekil 4.21.'e göre, Function kategorisindeki sınıflandırıcıların gerçek veri setlerinde benzer bir performans sergilediği görülebilir. Ancak ortalama ve medyanda en yüksek DSÖÖ değeri sağlayan sınıflandırıcı SimpleLogistic, en düşük DSÖO sağlayan sınıflandırıcının ise Logistic olduğu söylenebilir. Wilcoxon test sonuçları Çizelge 4.8.'de verilmiştir.

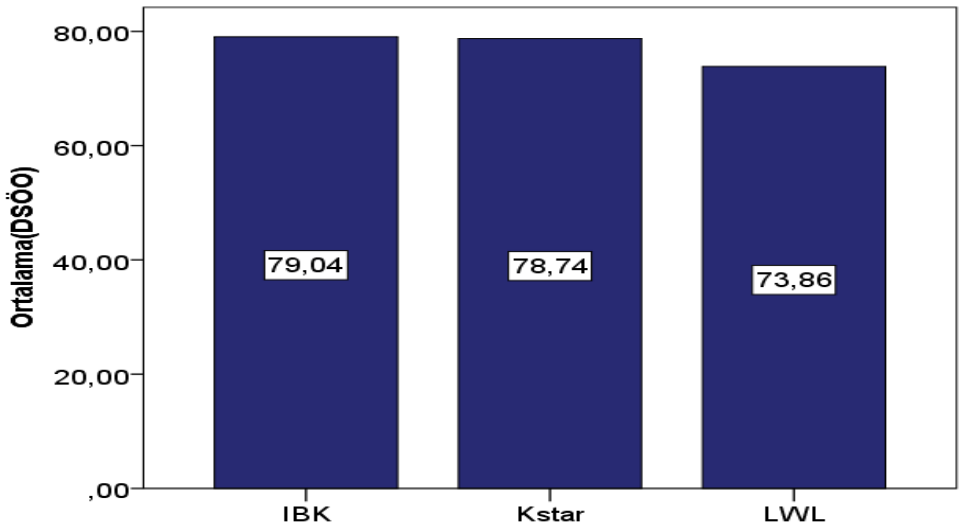
Çizelge 4.8. Gerçek veri setleri için Function sınıflandırıcılara ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
MultiLayerPerceptron-Logistic	37	49	0.071
SimpleLogistic-Logistic	36	47	0.017
SMO-Logistic	44	42	0.836
SimpleLogistic-MultiLayerPerceptron	34	50	0.213
SMO-MultiLayerPerceptron	47	37	0.203
SMO-SimpleLogistic	53	25	0.000

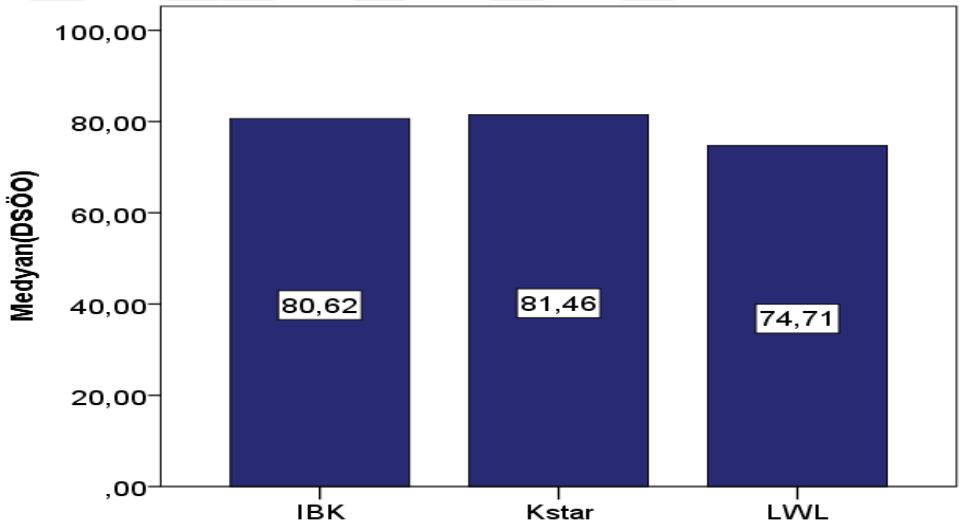
Çizelge 4.8.'e göre, SimpleLogistic-Logistic ve SMO-SimpleLogistic sınıflandırma çiftlerinin sınıflandırma başarılarının arasındaki fark istatistiksel açıdan anlamlı bulunmuştur. Diğer sınıflandırma çiftlerinin performanslarının benzer olduğu sonucuna ulaşılmıştır.

4.2.3 Gerçek veri setleri için Lazy kategorisi sınıflandırma sonuçları

Şekil 4.22. ve Şekil 4.23. Lazy kategorisindeki sınıflandırıcıların 100 gerçek veri setine uygulanması sonucunda elde edilen DSÖO değerlerinin ortalama ve medyan değerlerini göstermektedir.



Şekil 4.22. Gerçek veriler için Lazy sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.23. Gerçek veriler için Lazy sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.22. ve Şekil 4.23. incelendiğinde, ortalama değerlerine göre sınıflandırma başarısı en yüksek olan sınıflandırıcının IBk, medyan değerlerine göre ise Kstar yöntemi olduğu görülebilir. Bu kategoride sınıflandırma başarısı en düşük olan sınıflandırıcının LWL olduğu bulunmuştur. Wilcoxon test sonuçları ise Çizelge 4.9.'da verilmiştir.

Çizelge 4.9. Gerçek veri setleri için Lazy sınıflandırıcılara ilişkin Wilcoxon test sonuçları

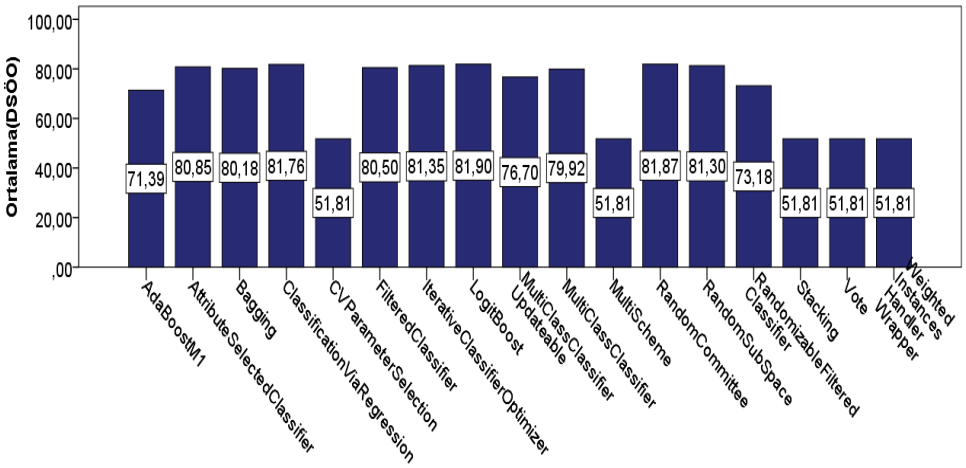
Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
Kstar-IBk	36	49	0.185
LWL-IBk	58	37	0.001

Çizelge 4.9.(Devam)

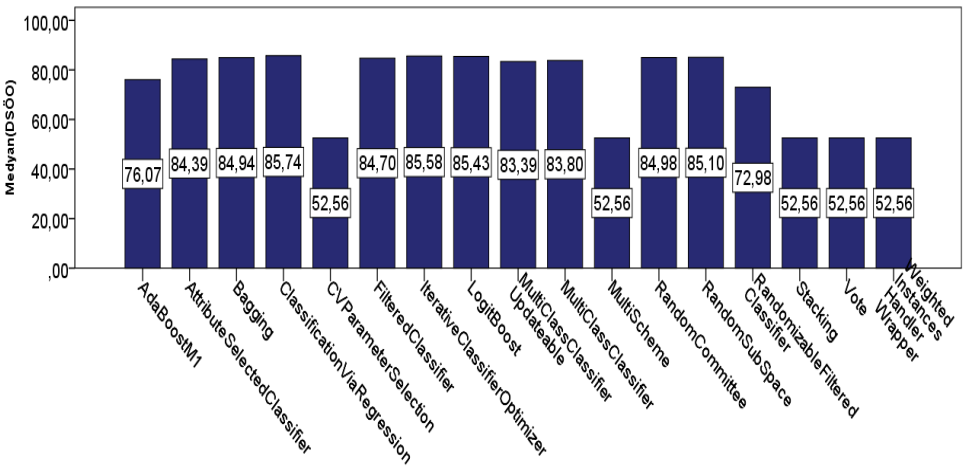
LWL-Kstar	60	34	0.001

Çizelge 4.9.'a göre Kstar ile IBk sınıflandırıcılarının performansları arasında anlamlı bir farklılık tespit edilememiştir. Ancak LWL sınıflandırıcısının IBk ve Kstar sınıflandırıcıları ile karşılaştırıldığında sınıflandırma başarısının anlamlı bir şekilde düşük olduğu söylenebilir.

4.2.4 Gerçek veri setleri için Meta kategorisi sınıflandırma sonuçları



Şekil 4.24. Gerçek veriler için Meta sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.25. Gerçek veriler için Meta sınıflandırıcılarına ait DSÖO medyan değerleri

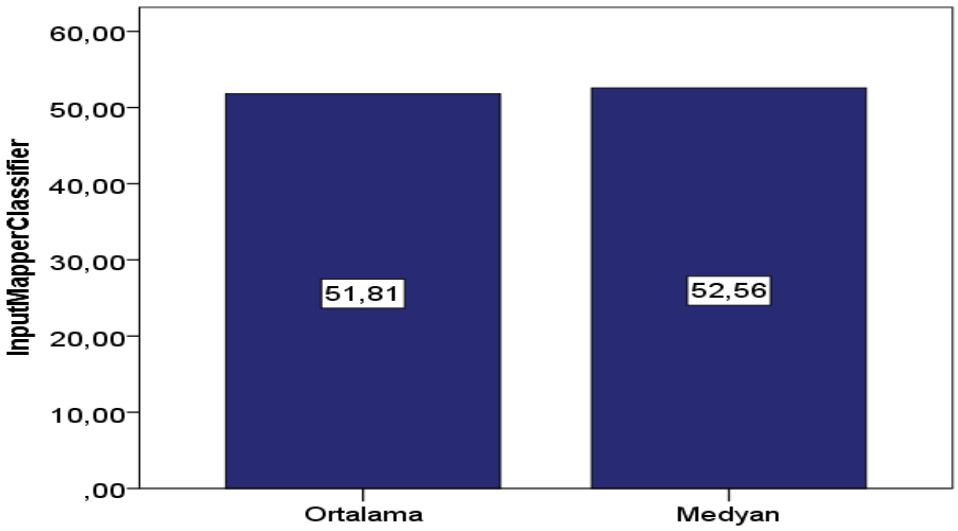
Şekil 4.24. ve Şekil 4.25.'e bakıldığında, simule edilmiş veriler ile benzer olarak CV Parameter Selection, Multi Scheme, Stacking, Vote, Weighted Instances Handler

Wrapper sınıflandırıcılarının tüm veri setlerinde aynı sonucu verdiği görülebilir. Bunun dışında, AttributeSelectedClassifier, Bagging, Classification Via Regression, Filtered Classifier, Iterative Classifier Optimizer, Logit Boost, Multi Class Classifier Updateable, Multi Class Classifier, Random Committee, Random Subspace sınıflandırıcıları %80'in üzerinde başarı sağlamıştır. Bu kategorideki sınıflandırıcılardan performanslarının istatistiksel olarak karşılaştırılmasına ilişkin test sonuçları EK-4'te verilmiştir. EK-4'ten elde edilen bazı önemli sonuçlar şu şekilde sıralanabilir.

- Ada Boost M1 sınıflandırıcısının performansı, CV Parameter Selection, Multi Scheme, Stacking, Vote, Weighted Instances Handler Wrapper sınıflandırıcılarından anlamlı bir şekilde yüksek, Randomizable Filtered Classifier ile benzer, diğer sınıflandırıcılardan anlamlı bir şekilde düşüktür.
- Attribute Selected Classifier ve Bagging sınıflandırıcıları, CV Paramater Selection, Multi Scheme, Randomizable Filtered Classifier, Stacking, Vote ve Weighted Instances Handler Wrapper sınıflandırıcılarından daha iyi bir performans sergilemiştir.
- Classification Via Regression sınıflandırıcısı, CV Parameter Selection, Filtered Classifier, Multi Class Classifier Updateable, Multi Class Classifier, Multi Scheme, Randomizable Filtered Classifier, Stacking, Vote ve Weighted Instances Handler Wrapper sınıflandırıcılarından, Filtered Classifier, Iterative Classifier Optimizer ve Logit Boost sınıflandırıcıları ise CV Parameter Selection, Multi Class Classifier Updateable, Multi Class Classifier, Multi Scheme, Randomizable Filtered Classifier, Stacking, Vote ve Weighted Instances Handler Wrapper daha yüksek bir sınıflandırma başarısı sağlamıştır.
- Filtered Classifier sınıflandırırıcısının sınıflandırma başarısı ile Iterative Classifier Optimizer ve Logit Boost sınıflandırıcılarının sınıflandırma başarıları arasındki fark istatistiksel açıdan anlamlıdır.

4.2.5 Gerçek veri setleri için Misc kategorisi sınıflandırma sonuçları

Şekil 4.26., Input Mapper Classifier sınıflandırıcsından elde edilen DSÖO değerlerinin ortalama ve medyan değerini göstermektedir.

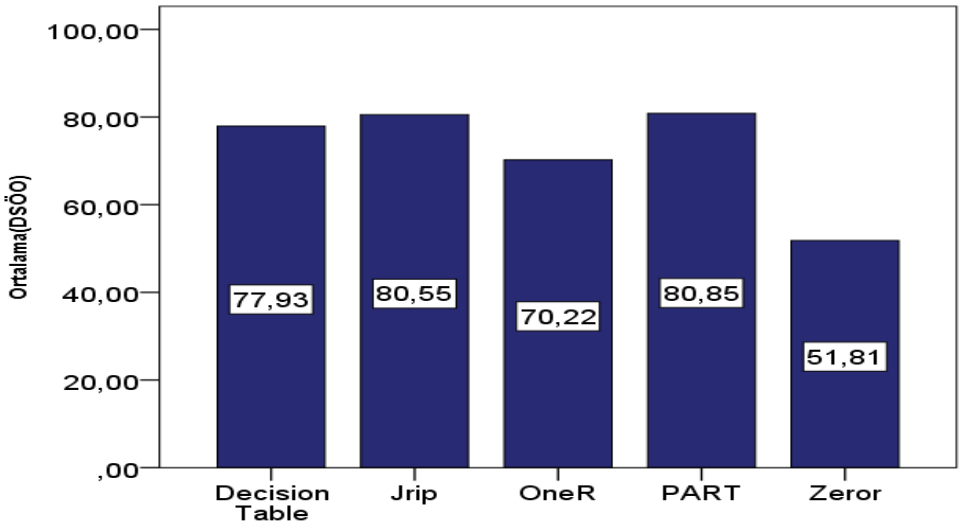


Şekil 4.26. Gerçek veriler için Misc sınıflandırıcılarına ait ortalama ve medyan DSÖO değerleri

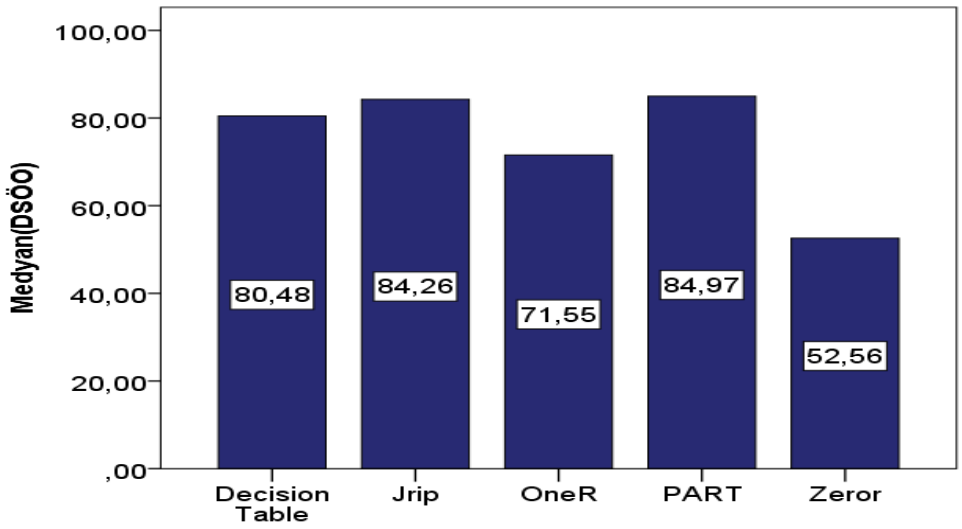
Şekil 4.26.'dan görüldüğü gibi Misc kategorisinden seçilen Input Mapper Classifier sınıflandırıcısı %50'ın üzerinde başarı sağlamıştır.

4.2.6 Gerçek veri setleri için Rules kategorisi sınıflandırma sonuçları

Rules kategorisine ait 5 sınıflandırma yönteminin ortalama ve medyan DSÖO değerleri sırasıyla Şekil 4.27. ve Şekil 4.28.'de verilmiştir.



Şekil 4.27. Gerçek veriler için Rules sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.28. Gerçek veriler için Rules sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.27. ile Şekil 4.28. incelendiğinde ortalama performansa göre en iyi sınıflandırıcıların JRip ve PART, medyan değerine göre PART olduğu görülmektedir. Bu kategoride en kötü sınıflandırma performansı ise ZeroR sınıflandırıcısından elde edilmiştir. Wilcoxon test sonuçları Çizelge 4.10.'da verilmiştir.

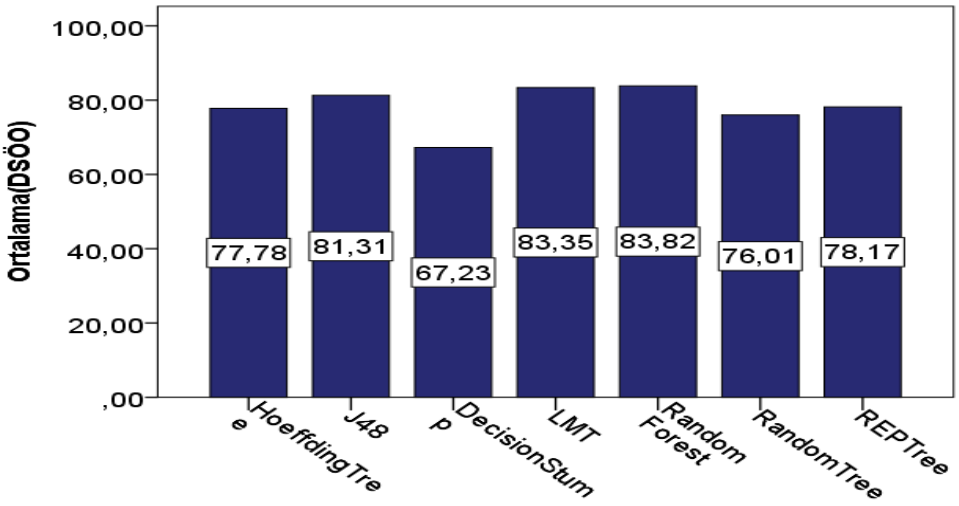
Çizelge 4.10. Gerçek veriler için Rules sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
JRip-DecisionTable	28	59	0.002
OneR-DecisionTable	70	14	0.000
PART-DecisionTable	30	56	0.000
ZeroR-DecisionTable	87	5	0.000
OneR-JRip	77	15	0.000
PART-JRip	30	53	0.121
ZeroR-JRip	83	9	0.000
PART-OneR	19	70	0.000
ZeroR-OneR	81	13	0.000
ZeroR-PART	84	11	0.000

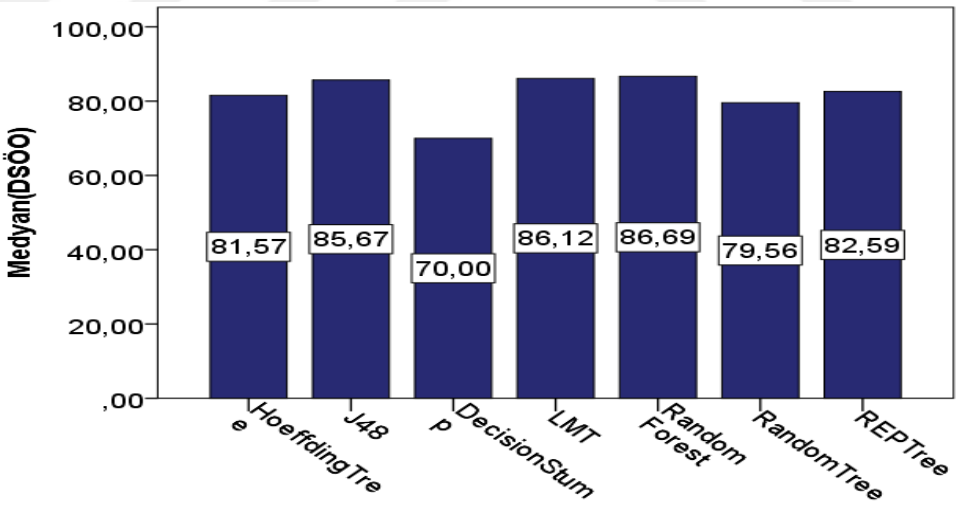
Çizelge 4.10.'a göre en iyi sınıflandırma başarısını veren PART ve JRip sınıflandırma yöntemlerinin performansları arasında anlamlı bir farklılık olmadığı söylenebilir. Ancak PART sınıflandırıcısının başarılı sonuçlar verdiği veri seti sayısı JRip yönteminden daha fazladır. Bunun dışında, diğer tüm sınıflandırma çiftlerinin performansları arasındaki fark istatistiksel açıdan anlamlı bulunmuştur.

4.2.7 Gerçek veri setleri için Trees kategorisi sınıflandırma sonuçları

Şekil 4.29. ve Şekil 4.30.'dan gerçek veri setleri için Trees kategorisindeki sınıflandırıcılardan elde edilen ortalama ve medyan DSSÖ değerlerini göstermektedir.



Şekil 4.29. Gerçek veri setlerinde Trees sınıflandırıcılarına ait DSÖO ortalama değerleri



Şekil 4.30. Gerçek veri setlerinde Trees sınıflandırıcılarına ait DSÖO medyan değerleri

Şekil 4.29. ve Şekil 4.30.'a bakıldığında, Trees kategorisindeki en başarılı sınıflandırıcıların LMT ve Random Forest, sınıflandırma açısından başarısı en düşük olan sınıflandırıcının Decision Stump olduğu görülebilir. Çizelge 4.11. Wilcoxon test sonuçlarını göstermektedir.

Çizelge 4.11. Gerçek veriler için Trees sınıflandırıcılarına ilişkin Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
J48-Hoeffding Tree	28	59	0.000
DecisionStump-HoeffdingTree	65	21	0.000
LMT-HoeffdingTree	17	68	0.000
RandomForest-HoeffdingTree	20	65	0.000
RandomTree-HoeffdingTree	57	41	0.042
REPTree-HoeffdingTree	36	50	0.270
DecisionStump-J48	74	14	0.000
LMT-J48	24	58	0.000
RandomForest-J48	23	60	0.000
RandomTree-J48	74	17	0.000
REPTree-J48	55	27	0.001
LMT-DecisionStump	9	81	0.000
RandomForest-DecisionStump	14	78	0.000
RandomTree-DecisionStump	39	57	0.000
REPTree-DecisionStump	21	64	0.000
RandomForest-LMT	32	46	0.329
RandomTree-LMT	85	13	0.000
REPTree-LMT	64	19	0.000
RandomTree-RandomForest	92	3	0.000
REPTree-RandomForest	75	11	0.000
REPTree-RandomTree	33	59	0.002

Çizelge 4.11.'e göre RandomForest ile LMT sınıflandırıcılarının performansları arasında anlamlı bir farklılık bulunamamıştır. Ancak Random Forest daha fazla veri seti için başarılı sonuç vermiştir. Bunun dışında, REP Tree-Hoeffding Tree sınıflandırma çiftinin performansı arasında da anlamlı bir farklılık tespit edilememiştir.

4.2.8 Gerçek veriler için genel karşılaştırma sonuçları

Bu bölümde, gerçek veri setleri için her bir sınıflandırma kategorisindeki en iyi performansa sahip sınıflandırma yöntemlerinin performansları karşılaştırılmıştır. Çizelge 4.12.'de en iyi performansa sahip sınıflandırma yöntemlerine ilişkin Wilcoxon test sonuçları verilmektedir.

Çizelge 4.12. Gerçek veriler ve en iyi sınıflandırıcılar için Wilcoxon test sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
SimpleLogistic-BayesNet	35	52	0.010
IBk-BayesNet	56	35	0.042
ClassificationViaRegression-BayesNet	32	52	0.015
PART-BayesNet	38	47	0.467
RandomForest-BayesNet	21	60	0.000
IBk-SimpleLogistic	63	22	0.000
ClassificationviaRegression-SimpleLogistic	45	37	0.785
PART-SimpleLogistic	46	33	0.043

Çizelge 4.12.(Devam)

RandomForest-SimpleLogistic	26	54	0.016
ClassificationViaRegression-IBk	26	60	0
PART-IBk	31	58	0.011
RandomForest-IBk	11	77	0
PART-ClassificationViaRegression	51	36	0.017
RandomForest-ClassificationViaRegression	24	59	0.001
RandomForest-PART	22	61	0

Çizelge 4.12.'ye bakıldığında, PART-Bayes Net ve Classification Via Regression-Simple Logistic sınıflandırma çiftlerinin performansları arasında anlamlı bir farklılık bulunamamıştır. Ortalamada en yüksek sınıflandırma başarısına sahip Random Forest sınıflandırma yönteminin performansının tüm sınıflandırıcılardan istatistiksel açıdan farklı olduğu görülmektedir. Buradan gerçek veri setlerinden en yüksek başarıyı sağlayan sınıflandırıcının RandomForest olduğunu söylemek mümkündür.

5. SONUÇ

Sınıflandırma, bir sınıf etiketine sahip olan ve çıktı değerleri belli olan veri seti üzerinden eğitilen bir modelin, sınıfı belli olmayan yeni bir gözlemin çıktı değerini tahmin edilmesinde kullanılmasıdır. Sınıflandırmanın geniş bir kullanım alanı vardır ve bu alanlar içerisinde hayati önem taşıyan sağlık sektörü ve tıp da bulunur. Bu sebeple makine öğrenmesi ile eğitilen sınıflandırıcıya ait modelin sınıflandırma işlemini en doğru şekilde yapması amaçlanmaktadır. Sınıflandırma için seçilebilecek farklı sınıflandırıcılar, farklı modeller oluştururlar ve yeni gözlem için farklı sınıf tahminleri yapabilme eğilimde olabilirler.

Literatürde bulunan çalışmalar özellikle belirli bir alana ait veri setleri ile ve belirli türdeki sınıflandırıcılar ile yürütülmüştür. Kısıtlı sayıdaki, aynı alana ait veri setleri ile birkaç sınıflandırıcı özelinde yapılan çalışmalar sayesinde elde edilen sonuçlar yine o alandaki veri setleri ve sınıflandırıcılar için örnek teşkil edebilmektedir. Bu test çalışmasında farklı tip ve büyüklükteki simule edilen ve gerçek veri setleri üzerinde farklı sınıflandırıcıların performansları karşılaştırılmıştır. Bunun için çevrimiçi veri depoları olan UCI ve Kaggle üzerinden sınıflandırma işlemi için uygun olan 100 adet veri seti elde edilmiş, sınıflandırma için ise açık kaynak kodlu veri madenciliği uygulaması olan WEKA'nın 3.8.3. versiyonu kullanılmıştır. Aynı zamanda simulasyon ile 100 adet veri seti üretilmiştir. Bu şekilde, 200 adet veri setine 41 adet sınıflandırıcı Hold-Out yöntemiyle uygulanmış, DSÖO açısından başarıları karşılaştırılmıştır.

Simulasyon veri setleri ile yürüttüğümüz çalışmada her sınıflandırıcı grubu için en başarılı ve başarısız olan sınıflandırıcılar ortalama ve medyan DSÖO değerleri açısından şu şekildedir:

Bayes kategorisinde, en yüksek başarıya sahip sınıflandırıcı Bayes Net olarak bulunmuştur. En düşük performansa sahip sınıflandırıcı ise Naive Bayes Multinomial Text sınıflandırıcısınındır.

Function kategorisinde, en düşük performansa sahip sınıflandırıcı SMO en yüksek performansa sahip sınıflandırıcı ise Multilayer Perceptron olarak bulunmuştur.

Lazy kategorisi için en iyi sınıflandırıcı IBk, en başarısız sınıflandırıcı LWL olarak bulunmuştur.

Meta kategorisindeki sınıflandırıcılar içerisinde CV Parameter Selection, Multi Scheme, Stacking, Vote ve Weighted Instances Handler Wrapper sınıflandırıcıları en düşük performansa, Random Committee ise en yüksek performansa sahip sınıflandırıcı olarak tespit edilmiştir.

Rules kategorisinde en düşük sınıflandırma başarısına sahip sınıflandırıcı ZeroR, en başarılı sınıflandırıcı ise PART olarak bulunmuştur.

Son olarak Trees kategorisindeki sınıflandırma yöntemleri arasında en başarılı sınıflandırıcı RandomForest, en düşük başarıya sahip sınıflandırıcı ise Hoeffding Tree olarak bulunmuştur.

Simulasyon ile yürütülen çalışmada genel olarak en başarısız algoritma %0 ortalama ile Naive Bayes Multinomial Text sınıflandırıcısı olmuştur. Genel kıyaslama sonucunda ise en başarılı sınıflandırıcı ise RandomForest olarak tespit edilmiştir.

Gerçek veri setleri ile yapılan karşılaştırmalar sonucunda ise aşağıdaki bulgular elde edilmiştir.

Bayes Kategorisinde, en başarılı sınıflandırıcı Bayes Net ve en başarısız sınıflandırıcı Naive Bayes Multinomial Text olarak bulunmuştur. Bu açıdan gerçek veri setleri ile simulasyon veri setlerinin en başarılı ve en başarısız sınıflandırıcıları aynıdır.

Function kategrisindeki en yüksek başarıya sahip sınıflandırıcı Simple Logistic, en düşük başarıya sahip sınıflandırıcı SMO'dur. Gerçek veri setleri ile simulasyon veri setleri en başarısız algoritma açısından örtüşseler de en başarılı algoritmaları farklılık göstermektedir.

Lazy kategorisinde en yüksek başarıya sahip sınıflandırıcı IBK iken en başarısız sınıflandıcırı LWL olmuştur. İki veri seti türü için de en başarılı ve en başarısız algoritmalar benzerlik göstermektedir.

Meta kategorisi için en başarılı algoritma Logit Boost olurken en başarısız algoritmalar aynı değere sahip olan CV Parameter Selection, Multi Scheme, Stacking, Vote, Weighted Instences Handler Wrapper olmuştur. En başarısız

algoritmalar her iki tür veri seti için de aynı olmasına rağmen en başarılı algoritma farklılık göstermektedir.

Rules kategorisinde en yüksek sınıflandırma başarısına sahip sınıflandırıcı PART olurken en başarısız sınıflandırıcı ZeroR olarak bulunmuştur. Hem gerçek hem de simulasyon veri setleri için en başarılı ve en başarısız sınıflandırıcılar örtüşmektedir.

Son olarak ise Trees kategorisi içerisinde en başarılı sınıflandırıcı Random Forest iken en başarısız sınıflandırıcı Decision Stump olmuştur. En başarılı algoritmalar iki tür veri seti için de örtüşse de en başarısız algoritmalar farklıdır.

Gerçek veri setleri içerisinde en düşük değeri %51.8 olarak CV Parameter Selection, Input Mapped Classifier, Multi Scheme, Stacking, Vote, Weighted Instances Handler Wrapper ve ZeroR sınıflandırıcılarında gözlemlenmiştir. En yüksek değere sahip sınıflandırıcı ise %83.8 ile Random Forest olmuştur.

Hem gerçek veri setleri, hem de simulasyon ile elde edilen veri setleri için ortalama DSÖO açısından en yüksek başarıyı gösteren ilk 10 sınıflandırma algoritması Çizelge 5.1. içerisinde verilmiştir.

Çizelge 5.1. Gerçek veri setleri ve simulasyon veri setleri için en başarılı 10 sınıflandırma algoritması ve DSÖO değerleri

Gerçek Veri Setleri			Simulasyon Veri Setleri		
Sıra	Sınıflandırma Algoritması	DSÖO	Sıra	Sınıflandırma Algoritması	DSÖO
1	Random Forest	83.816673	1	Random Forest	81.693481
2	LMT	83.353664	2	Random Committee	80.247308
3	Logit Boost	81.901595	3	LMT	78.710592
4	Random Committee	81.869036	4	Classification Via Regression	78.384692
5	Classification Via Regression	81.758797	5	PART	77.715809
6	Simple Logistic	81.732145	6	Bagging	77.64565
7	Iterative Classifier Optimizer	81.354762	7	J48	76.828421
8	J48	81.3095	8	Logit Boost	76.412094
9	Random Sub Space	81.304028	9	Iterative Classifier Optimizer	76.22629
10	PART	80.848885	10	Random Sub Space	75.001645

Tablo 5.1. bize göstermekte ki her iki tip veri setleri içerisinde Random Forest algoritması en yüksek başarıyı göstermektedir. Ayrıca DSÖO incelendiğinde gerçek veri setleri ile çalışmak, simulasyon veri setlerine göre daha başarılı sonuçlar elde edilmesini sağlamaktadır. Örneğin gerçek veri setlerinin başarı listesinde 10. sırada

olan PART algoritması yaklaşık olarak %80.8 başarı göstermiştir. Simulasyon veri setleri içerisindeki 5. başarı sırasında yer alan PART algoritmasının başarısı ise %77.7 olarak elde edilmiştir.

Yapılan çalışmada No Free Lunch teoremini destekleyen gözlemler yapılmış, hiçbir sınıflandırıcının her veri seti üzerinde mutlak en iyi başarıyı sergilemediği görülmüş, veri madencilerinin farklı sınıflandırıcıları denemeleri gerektiği belirlenmiştir. Ancak bu denemeler için ağaç tabanlı bir algoritma olan Random Forest, karara ağaçları ve lojistik regresyonun birleşiminden oluşan LMT, lojistik regresyon temelli olan Logit Boost sınıflandırma algoritmalarının ilk tercih sebebi olması gerektiğini görmüş oluyoruz.

KAYNAKLAR

Abdeldaim, A.M., Sahlol, A.T., Elhoseny, M. ve Hassanien, A.E. (2018) Computer-Aided Acute Lymphoblastic Leukemia Diagnosis System Based on Image Analysis, 131-147, Hassanien, A.E. ve Oliva, D.A. (editörler), *Advances in Soft Computing and Machine Learning in Image Processing*, Springer, Switzerland, 718

Aggarwal, C.C. ve Zhai, C.X. (2012) *An Introduction to Text Mining*, Springer Science+Business Media, New York, 522s.

Agraval, R.K. ve Bala, R. (2008) Incremental Bayesian Classification for Multivariate Normal Distribution Data, *Pattern Recognition Letters*, 29: 1873-1876

Ahmed, S.R. (2004) Applications of Data Mining in Retail Business, *International Conference on Information Technology: Coding and Computing*, 5-7 Nisan 2004, Las Vegas, USA, 455-459

Ali, S. ve Smith, K.A. (2006) On Learning Algorithm Selection for Classification, *Applied Soft Computing*, 6(2): 119-138

Al-Taie, M.Z., Kadry, S. ve Lucas, J.P. (2019) Online Data Preprocessing: A Case Study Approach, International *Journal of Electrical and Computer Engineering(IJECE)*, 9(4): 2620-2626

Amala, G. (2019) Orange Tool Approach for Comporative Analysis of Supervised Learning Algorithm in Classification Mining, *Journal of Analysis and Computation(JAC)*, 13(1): 1-10

Basha, G. (2017) Importance of Data Mining in Banking Sectors, *Intenational Journal of Scientific Engineering and Technology Research*, 6(7): 1264-1267

Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meini, T., Ohl, P., Thiel, K. ve Wiswedel, B. (2009) KNIME – The Konstanz Information Miner, ACM SIGKDD Explorations Newsletter, 11(1): 26-31

Bhargava, N., Sharma, G., Bhargava, R. ve Mathuria, M. (2013) Decision Tree Analysis on J48 Algorithm for Data Mining, *IJARCSSE,* 3(6): 1114-1119

Bienvenido-Huertas, D., Nieto-Julián, J.E., Moyano, J.J., Macías-Bernal, J.M. ve Castro, J. (2019) Implementing Artificial Intelligence in H-BIM Using the J48 Algorithm to Manage Historic Buildings, *International Journal of Architectural Heritage*(2019), 1-13

Boser, B.E., Guyon, I.M. ve Vapnik V.N. (1992) A Training Algorithm for Optimal Margin Classifiers, Proc. 5th. ACM Workshop on Computational Learning Theory (COLT), Pittsburg, Pennsylvania, USA, 144-152

Cao, X.H., Stajkovic, I. ve Obradovic Z. (2016) A Robust Data Sclaing Algorithm to Improve Classification Accuracies in Biomedical Data, *BMC Bioinformatics,* 17: 359-368

Chandaka, S., Chatterjee, A. ve Munshi, S. (2009) Cross-Correlation Aided Support Vector Machine Classifier for Classification of EEG Signals, *Expert Systems with Applications*, 36: 1329-1336

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

Chen, S., Goo,Y.-J.J. ve Shen, Z,-D. (2014) A Hybrid Approach of Stepwise Regression, Logistic Regression, Support Vector Machine, and Decision Tree for Forecasting Fraudulent Financial Statements, *The Scientific World Journal*, 2014: 9 pages

Chen, Y. (2020) Automatic Microseismic Event Picking via Unsupervised Machine Learning, *Geophysical Journal International*, 222(3): 1750-1764

Dash, R., Paramguru, R.L. ve Dash, R. (2011) Comparative Analysis of Supervised and Unsupervised Discretization Techniques, *International Journal of Advances in Science and Technology*, 2(3): 29-37

Dwivedi, S., Kasliwal, P. ve Soni, S. (2016) Comprehensive Study of Data Analytics Tools (RapidMiner, Weka, R Tool, Knime), *Symposium on Colossal Data Analysis and Networking*, 18-19 Mart 2016, Indore, Hindistan, 1-8

Everingham, Y., Sexton, J., Skocaj, D. ve Inman-Bamber, G. (2016) Accurate Prediction of Sugarcane Yield Using a Random Forest Algorithm, *Agronomy for Sustainable Development*, 36(2): 1-9

Ferri, C., Hernández-Orallo, J. ve Modroiu, R. (2009) An Experimental Comparison of Performance Measures for Classification, *Pattern Recognition Letters*, 30(1): 27-38

Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B. ve Witten, I.H.(2010) WEKA-A Machine Learning Workbench for Data Mining, 1269-1277, Maimon, O. ve Rokach, L. (editörler), *Data Mining and Knowledge Discovery Handbook*, 2. Baskı, Springer Science+Business Media, New York, 1285s.

García, S., Luengo, J. ve Herrera, F. (2015) *Data Preprocessing in Data Mining*, 72, Springer, Switzerland, 320

Gullo, F. (2015) From Patterns in Data to Knowledge Discovery: What Data Mining Can Do, *3rd International Conference Frontiers in Diagnostic Technologies*, 25-27 Kasım, İtalya, 18-22

Hamoud, A.K., Majeed, A., Awadh, W.A. ve Hashim, A.S. (2017) Students' Success Prediction Based on Bayes Algorithms, *International Journal of Computer Applications*, 178(7): 6-12

Han, J., Kamber, M. ve Pei, J. (2011) *Data Mining: Concepts and Techniques*, 3. Baskı, Elsevier Inc., Waltham, U.S.A., 744s.

Hassan, C.A.U., Khan, M.S. ve Shah, M.A. (2018) Comparison of Machine Learning Algorithms in Data Classification, *International Conference on Automation and Computing*, 6-7 Eylül 2018, Newcastle/United Kingdom, 1-6

Huang, Z. ve Liang, Y. (2019) Research of Data Mining and Web Technology in University Discipline Construction Decision Support System Based on MVC Model, *Library Hi Tech*, 38(3): 610-624

Islam, R. ve Xiang, Y. (2010) Email Classification Using Data Reduction Method, *Proceedings of the 5th International ICST Conference on Communications and Networking in China*, 16 Haziran 2010, Piscataway, 1-5

Jony, R.I., Mohammed, N., Habib, A., Momen, S. ve Rony, R.I. (2015) An Evaluation of Data Processing Solutions Considering Preprocessing and "Special" Features, *11th International Conference on Signal-Image Technology & Internet-Based Systems*, Kasım 2015, Bangkok Thailand, 224-231

Kalpana, D. (2017) Data Mining Apriori Algorithm Implementation Using R, Big Data Analystics Using R, *International Research Journal of Engineering and Technology(IRJET)*, 4(11): 1810-1815

Kaur, G. ve Chhabra, A. (2014) Improved J48 Classification Algorithm fort he Prediction of Diabetes, *International Journal of Computer Applications*, 98(22): 13-17

Kaur, M. ve Kang, S. (2016) Market Basket Analysis: Identify the Changing Trends of Market Data Using Association Rule Mining, *International Conference on Computational Modeling and Security*, Şubat 2016, Hindistan, 85: 78-85

Koh, H.C. ve Tan, G. (2011) Data Mining Applications in Healthcare, *Journal of Healthcare Information Management*, 19(2): 64-72

Kulkarni, A.D. ve Shrestha, A. (2017) Multispectral Image Analysis Using Decision Trees, International Journal of Advanced Computer Science and Applications, 8(6): 11-18

Li, H., Yuan, D., Ma, X., Cui, D. ve Cao, L. (2017) Genetic Algorithm for the Optimization of Features and Neural Networks in ECG Signals Classification, Scientific Reports, 7(1): 1-12

Longadge, R., Dongre, S.S. ve Malik, L. (2013) Class Imbalance Problem in Data Mining: Review, IJCSN, 2(1): 83-87

Mhetre, V. ve Nagar, M. (2017) Classification Based Data Mining Algorithms to Predict Slow, Average and Fast Learners in Educational System Using WEKA, International *Conference on Computing Methodologies and Communication*, 2017, Mumbai, 475-479

Moavenian, M. ve Khorrami, H. (2010) A Qualitative Comparison of Artificial Neural Networks and Support Vector Machines ECG Arrhythmias Classification, *Expert Systems With Applications*, 37(4): 3088-3093

Naik, A. ve Samant, L. (2016) Correlation Review of Classification Algorithm Using Data Mining Tool: WEKA, Rapidminer, Tanagra, Orange and Knime, *International Conference on Computational Modeling and Security(CMS)*, 13 Şubat 2016, Bangalore, India, 85: 662-668

Nookala, G.K.M., Pottumuthu, B.K., Orsu, N. ve Mudunuri, S.B. (2013) Performance Analysis and Evaluation of Different Data Mining Algorithms used for Cancer Classification, International Journal of Advanced Research in Artificial Intelligence, 2(5): 49-55

Olson, D.L. ve Delen, D. (2008) *Advanved Data Mining Techniques*, Springer, Berlin, Almanya, 180s.

Omary, Z. ve Mtenzi, F. (2009) Dataset Threshold for the Performance Estimators in Supervised Machine Learning Experiments, International Conference for Internet Technology and Secured Transactions, 9-13 Kasım 2009, Londra, Birleşik Krallık, 1-8

Osisanwo, F.Y., Akinsolo, J.E.T., Awodele, O., Hinmikaiye, J.O., Olakanmi, O. ve Akinjobi, J. (2017) Supervised Machine Learning Algorithms: Classification and Comparison, *International Journal of Computer Trens and Technology*, 48(3): 128-138

Pandey, U.K. ve Pal, S. (2011) Data Mining: A Prediction of Performer or Underperformer Using Classification, *IJCSIT*, 2(2): 686-690

Patil, S. (2016) Big Data Analystics Using R, *International Research Journal of Engineering and Technology(IRJET)*, 3(7): 78-81

Rashidi, H.H., Tran, N.K., Betts, E.V., Howell, L.P. ve Grenn, R. (2019) Artificial Intelligence and Machine Learning in Pathology: The Present Lanscape of Supervised Methods, Academic Pathology, 6: 1-17

Saigal, P., Chandra, S. ve Rastogi, R. (2019) Multi-Category Ternion Support Vector Machine, Engineering Applications of Artificial Intelligence, 85: 229-242

Shi, G. (2014) *Data Mining and Knowledge Discovery for Geoscientists*, Elseiver Inc., Waltham, U.S.A, 376s.

Sim, D.Y.Y., Teh, C.S. ve Ismail, A.I. (2017) Improved Boosting Algorithms by Pre-Pruning and Associative Rule Mining on Decision Trees for Predicting Obstructive Sleep Apnea, *Advanced Science Letters*, 4(1): 401-407

Soiraya, M., Thanalerdmongkol, S. ve Chantrapornchai, C. (2012) Using a Data Mining Approach: Spam Detection on Facebook, *International Journal of Computer Applications*, 58(13): 26-31

Venkatadri, M. ve Reddy, L.C. (2011) A Review on Data Mining from Past to Future, *International Journal of Compıter Applications*, 15(7): 19-22

Verma, D. ve Mishra, N. (2017) Analysis and Prediction of Breast Cancer and Diabetes Disease Datasets Using Data Mining Classification Techniques, *International Conference on Intelligent Sustainable Systems*, Tiruppur, Hindistan, 7-8 Aralık 2017, 533-538

Wu, Z., Xu, Q., Li, J., Fu, C., Xuan, Q. ve Xiang, Y. (2017) Passive Indoor Localization Based on CSI and Naive Bayes Classification, *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 48(9): 1566-1577

Yılmaz, M. (2018) *Tarımsal Yaz Ürünlerin Sentinel-2 Uydu Görüntülerinden Rastgele Orman Algoritması ile Nesne-Tabanlı Sınıflandırılması*, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara, 78

Zhang, Y, Guo, S.L., Han, L.N. ve Li, T.L. (2016) Application and Exploration of Big Data Mining in Clinical Medicine, *Chinese Medical Journal*, 129(6):731-738

Zhao, Y. ve Zhang, Y. (2008) Comparison of Decision Tree Methods for Finding Active Objects, Advances in Space Research, 41: 1955-1959

Zhongguo, Y., Hongqi, L., Ali, S. ve Yile, A. (2017) Choosing Classification Algorithms and Its Optimum Parameters Based on Data Set Characteristics, *Journal of Computers*, 28(5): 26-38

URL1: https://www.uib.no/en/rg/ml/119695/bayesian-networks

EKLER

EK-A: Simule Edilmiş Veri Setlerinin Özellikleri

	Gözlem Sayısı	Değişken Sayısı	Sınıf Sayısı
a1	600	8	7
a2	100	10	2
a3	1000	6	5
a4	1000	21	10
a5	250	11	6
a6	2000	6	7
a7	1000	13	20
a8	1000	11	5
a9	50	6	3
a10	50	6	3
a11	50	6	3
a12	500	11	6
a13	250	11	3
a14	400	11	4
a15	500	11	6
a16	100	10	2
a17	600	6	6
a18	500	10	2
a19	555	10	10
a20	250	25	10
a21	500	25	10
a22	100	25	10
a23	650	25	10
a24	300	7	4
a25	1250	6	11
a26	600	16	4
a27	600	16	4
a28	1250	6	11
a29	600	8	7
a30	2500	7	14
a31	50	6	3
a32	450	8	4
a33	775	8	7
a34	146	9	7
a35	340	4	5
a36	2000	4	6
a37	226	4	3
a38	1230	5	4
a39	150	3	1
a40	2250	7	6
a41	450	10	6
a42	2250	10	3
a43	250	10	3

EK-A.(Devam)

a44	600	6	4
a45	750	19	4
a46	321	6	6
a47	524	10	4
a48	550	8	5
a49	324	6	9
a50	550	8	5
a51	1000	11	5
a52	1000	11	5
a53	555	12	6
a54	1000	10	2
a55	1521	8	9
a56	750	7	8
a57	1500	13	4
a58	1901	4	6
a59	428	10	6
a60	428	10	6
a61	500	10	2
a62	500	10	2
a63	1000	11	5
a64	1000	6	4
a65	600	8	6
a66	300	11	3
a67	450	8	4
a68	1234	6	7
a69	1000	4	3
a70	1000	4	5
a71	550	6	7
a72	550	6	7
a73	350	6	3
a74	350	6	3
a75	350	6	3
a76	350	6	3
a77	450	6	19
a78	450	6	11
a79	234	7	2
a80	36	6	3
a81	1000	10	6
a82	250	6	5
a83	250	6	5
a84	650	11	7
a85	250	11	4
a86	850	4	6
a87	850	5	4
a88	850	5	4
a89	50	5	4
a90	150	5	4
a91	350	5	5
a92	125	5	3
a93	250	25	10
a94	300	11	3

EK-A.(Devam)

a95	2341	9	3
a96	3000	5	8
a97	356	4	3
a98	1356	5	4
a99	135	6	5
a100	352	7	6

EK-B: Simule Edilmiş Veriler İçin Meta Sınıflandırıcılara İlişkin Wilcoxon Test Sonuçları

Sınıflandırma Çifti	**Negatif Rank**	**Pozitif Rank**	**p**
AttributeSelectedClassifier-AdaBoostM1	8	86	0
Bagging-AdaBoostM1	6	87	0
ClassificationViaRegression-AdaBoostM1	2	94	0
CVParameterSelection-AdaBoostM1	86	6	0
FilteredClassifier-AdaBoostM1	10	82	0
IterativeClassifierOptimizer-AdaBoostM1	3	90	0
LogitBoost-AdaBoostM1	2	92	0
MultiClassClassifierUpdateable-AdaBoostM1	36	56	0
MultiClassClassifier-AdaBoostM1	17	87	0
MultiScheme-AdaBoostM1	86	6	0
RandomCommittee-AdaBoostM1	6	93	0
RandomSubSpace-AdaBoostM1	11	83	0
RandomizableFilteredClassifier-AdaBoostM1	29	69	0
Stacking-AdaBoostM1	86	6	0
Vote-AdaBoostM1	86	6	0
WeightedInstancesHandlerWrapper-AdaBoostM1	86	6	0
Bagging-AttributeSelectedClassifier	19	74	0
ClassificationViaRegression-AttributeSelectedClassifier	15	73	0
CVParameterSelection-AttributeSelectedClassifier	93	1	0
FilteredClassifier-AttributeSelectedClassifier	57	25	0
IterativeClassifierOptimizer-AttributeSelectedClassifier	30	58	0.007

EK-B(Devam)

LogitBoost-AttributeSelectedClassifier	25	66	0.001
MultiClassClassifierUpdateable-AttributeSelectedClassifier	86	9	0
MultiClassClassifier-AttributeSelectedClassifier	76	18	0
MultiScheme-AttributeSelectedClassifier	93	1	0
RandomCommittee-AttributeSelectedClassifier	14	80	0
RandomSubSpace-AttributeSelectedClassifier	39	58	0.13
RandomizableFilteredClassifier-AttributeSelectedClassifier	46	52	0.115
Stacking-AttributeSelectedClassifier	93	1	0
Vote-AttributeSelectedClassifier	93	1	0
WeightedInstancesHandlerWrapper-AttributeSelectedClassifier	93	1	0
ClassificationViaRegression-Bagging	51	40	0.719
CVParameterSelection-Bagging	96	3	0
FilteredClassifier-Bagging	79	10	0
IterativeClassifierOptimizer-Bagging	57	29	0.01
LogitBoost-Bagging	58	31	0
MultiClassClassifierUpdateable-Bagging	91	7	0
MultiClassClassifier-Bagging	86	12	0
MultiScheme-Bagging	96	3	0
RandomCommittee-Bagging	27	64	0
RandomSubSpace-Bagging	60	29	0
RandomizableFilteredClassifier-Bagging	58	40	0.002
Stacking-Bagging	96	3	0
Vote-Bagging	96	3	0
WeightedInstancesHandlerWrapper-Bagging	96	3	0
CVParameterSelection-ClassificationViaRegression	97	2	0
FilteredClassifier-ClassificationViaRegression	83	10	0
IterativeClassifierOptimizer-ClassificationViaRegression	49	37	0.003
LogitBoost-ClassificationViaRegression	53	36	0.002
MultiClassClassifierUpdateable-ClassificationViaRegression	91	4	0
MultiClassClassifier-ClassificationViaRegression	92	5	0

EK-B(Devam)

MultiScheme-ClassificationViaRegression	97	2	0
RandomCommittee-ClassificationViaRegression	27	66	0
RandomSubSpace-ClassificationViaRegression	63	33	0
RandomizableFilteredClassifier-ClassificationViaRegression	54	44	0
Stacking-ClassificationViaRegression	97	2	0
Vote-ClassificationViaRegression	97	2	0
WeightedInstancesHandlerWrapper-ClassificationViaRegression	97	2	0
FilteredClassifier-CVParameterSelection	1	91	0
IterativeClassifierOptimizer-CVParameterSelection	2	96	0
LogitBoost-CVParameterSelection	1	98	0
MultiClassClassifierUpdateable-CVParameterSelection	3	79	0
MultiClassClassifier-CVParameterSelection	5	93	0
MultiScheme-CVParameterSelection	0	0	1
RandomCommittee-CVParameterSelection	0	99	0
RandomSubSpace-CVParameterSelection	0	96	0
RandomizableFilteredClassifier-CVParameterSelection	6	92	0
Stacking-CVParameterSelection	0	0	1
Vote-CVParameterSelection	0	0	1
WeightedInstancesHandlerWrapper-CVParameterSelection	0	0	1
IterativeClassifierOptimizer-FilteredClassifier	19	71	0
LogitBoost-FilteredClassifier	17	74	0
MultiClassClassifierUpdateable-FilteredClassifier	84	9	0
MultiClassClassifier-FilteredClassifier	71	26	0
MultiScheme-FilteredClassifier	91	1	0
RandomCommittee-FilteredClassifier	11	86	0
RandomSubSpace-FilteredClassifier	28	70	0.001
RandomizableFilteredClassifier-FilteredClassifier	44	55	0.56
Stacking-FilteredClassifier	91	1	0
Vote-FilteredClassifier	91	1	0
WeightedInstancesHandlerWrapper-FilteredClassifier	91	1	0
LogitBoost-IterativeClassifierOptimizer	22	18	0.633

EK-B(Devam)

MultiClassClassifierUpdateable-IterativeClassifierOptimizer	90	6	0
MultiClassClassifier-IterativeClassifierOptimizer	87	11	0
MultiScheme-IterativeClassifierOptimizer	96	2	0
RandomCommittee-IterativeClassifierOptimizer	26	68	0
RandomSubSpace-IterativeClassifierOptimizer	48	44	0.436
RandomizableIterativeClassifierOptimizer-IterativeClassifierOptimizer	49	49	0.018
Stacking-IterativeClassifierOptimizer	96	2	0
Vote-IterativeClassifierOptimizer	96	2	0
WeightedInstancesHandlerWrapper-IterativeClassifierOptimizer	96	2	0
MultiClassClassifierUpdateable-LogitBoost	92	7	0
MultiClassClassifier-LogitBoost	84	7	0
MultiScheme-LogitBoost	98	1	0
RandomCommittee-LogitBoost	26	67	0
RandomSubSpace-LogitBoost	50	44	0.391
RandomizableFilteredClassifier-LogitBoost	49	48	0.011
Stacking-LogitBoost	98	1	0
Vote-LogitBoost	98	1	0
WeightedInstancesHandlerWrapper-LogitBoost	98	1	0
MultiClassClassifier-MultiClassClassifierUpdateable	14	82	0
MultiScheme-MultiClassClassifierUpdateable	79	3	0
RandomCommittee-MultiClassClassifierUpdateable	2	96	0
RandomSubSpace-MultiClassClassifierUpdateable	10	88	0
RandomizableFilteredClassifier-MultiClassClassifierUpdateable	32	65	0
Stacking-MultiClassClassifierUpdateable	79	3	0
Vote-MultiClassClassifierUpdateable	79	3	0
WeightedInstancesHandlerWrapper-MultiClassClassifierUpdateable	79	3	0
MultiScheme-MultiClassClassifier	93	5	0
RandomCommittee-MultiClassClassifier	4	92	0
RandomSubSpace-MultiClassClassifier	18	79	0
RandomizableFilteredClassifier-MultiClassClassifier	41	58	0.019
Stacking-MultiClassClassifier	93	5	0
Vote-MultiClassClassifier	93	5	0
WeightedInstancesHandlerWrapper-MultiClassClassifier	93	5	0
RandomCommittee- MultiScheme	0	99	0
RandomSubSpace- MultiScheme	0	96	0

EK-B(Devam)

RandomizableFilteredClassifier-MultiScheme	6	92	0
Stacking- MultiScheme	0	0	1
Vote- MultiScheme	0	0	1
WeightedInstancesHandlerWrapper-MultiScheme	0	0	1
RandomSubSpace- RandomCommittee	83	12	0
RandomizableFilteredClassifier-RandomCommittee	58	40	0
Stacking- RandomCommittee	99	0	0
Vote- RandomCommittee	99	0	0
WeightedInstancesHandlerWrapper-RandomCommittee	99	0	0
RandomizableFilteredClassifier-RandomSubSpace	47	49	0.059
Stacking- RandomSubSpace	96	0	0
Vote- RandomSubSpace	96	0	0
WeightedInstancesHandlerWrapper-RandomSubSpace	96	0	0
Stacking- RandomizableFilteredClassifie	92	6	0
Vote- RandomizableFilteredClassifier	92	6	0
WeightedInstancesHandlerWrapper-RandomizableFilteredClassifier	92	6	0
Vote- Stacking	0	0	1
WeightedInstancesHandlerWrapper-Stacking	0	0	1
WeightedInstancesHandlerWrapper- Vote	0	0	1

EK-C: Gerçek Veri Setlerinin Özellikleri

Veri Seti	**n**	**Öznitelik Sayısı**	**Sınıf Sayısı**
Acute Inflammations	120	7	2
Adult	32560	15	2
Audiology(Standardized)	200	71	24
Autism Screening Adult	704	21	2
Autistic Spectrum Disorder Screenning Data For Children	292	21	2
Avila	20867	11	12
Balance Scale	625	5	3
Balloons	20	5	2
Banknote Authentication	1372	5	2
Blood Transfusion Service Center	748	5	2
Breast Cancer	286	10	2
Breast Cancer Coimbra	116	10	2
Breast Cancer Wisconsin(Diagnostic)	699	11	2
Burst Header Packet(BHP) Flooding Attack on Optical Burst Switching(OBS) Network	1075	22	4
Ceaserian Section	80	6	2
Car Evaluation	1728	7	4
Census Income	32561	15	2
Cervical Cancer(Risk Factors)	858	36	2
Chess (King-Rook vs. King-Pawn)	3196	37	2
Chronic_Kidney_Disease	400	25	2
Congressional Voting Records	435	17	2
Connectionist Bench (Sonar, Mines vs. Rocks)	208	61	2
Contraceptive Method Choice	1473	10	3
Credit Approval	690	16	2
Crowdsourced Mapping	10545	29	6
Cryotherapy	90	7	3
Cylinder Bands	540	40	2
Dermatology	366	35	6
Diabetes	768	9	2
Diabetic Retinopathy Debrecen	1151	20	2
Dresses Attribute Sales	500	14	2
Ecoli	336	9	8
EEG Eye State	14980	15	2
Electrical Grid Stability Simulated	10000	14	2
extention of Z-Alizadeh sani	303	11	2
Fertility	100	10	2
Flags	194	30	8
Forest Type Mapping	523	28	4
Glass Identification	214	11	6
Haberman's Survival	306	4	2
Hayes-Roth	132	6	3
HCC Survival	165	50	2
Hepatitis	155	20	2
Hill-Valley	1212	101	2
Horse Colic	300	28	2
HTRU2	17898	9	2
LPD (Indian Liver Patient Dataset)	583	11	2

EK-C(Devam)

Image Segmentation	210	20	7
Immunotherapy	90	8	2
Ionosphere	351	35	2
Iris	150	5	3
Japanese Credit Screening	690	16	2
Leaf	340	16	30
Lenses	24	5	3
Letter Recognition	20000	17	26
Lung Cancer	32	57	3
Lymphography	148	19	4
MAGIC Gamma Telescope	19020	11	2
Mammographic Mass	961	6	2
Mice Protein Expression	1080	82	8
Mushroom	8124	23	2
Nursery	12960	9	5
Phishing Websites	11055	31	2
Post-Operative Patient	90	9	4
Primary Tumor	339	18	21
QSAR biodegradation	1055	42	2
Qualitative_Bankruptcy	250	7	2
Scadi	70	206	7
Seeds	210	8	3
Seismic-Bumps	2584	19	2
Shuttle Landing Control	15	7	2
Somerville Happiness Survey	143	7	2
Soybean(Large)	307	36	19
Spambase	4601	58	2
Statlog(Australian Credit Approval)	690	15	2
Statlog(German Credit Data)	1000	21	2
Statlog(Heart)	270	14	2
Statlog(Shuttle)	14500	10	7
Student Academics Performance	131	22	3
Teaching Assistant Evaluation	151	6	3
Thoracic Surgery	470	17	2
Tic-tac-toe Endgame	958	10	2
User Knowledge Modeling	258	6	4
Wall-Following Robot Navigation	5456	25	4
Waveform Database Generator	5000	22	3
Website Phishing	1353	10	3
Wholesale Customers	440	8	2
Wine	178	14	3
Wine Quality(Red)	1599	12	6
Wine Quality(White)	4898	12	7
Wireless Indoor Localization	2000	8	4
Yeast	1484	9	10
Z-Alizadeh Sani	303	56	2
Zoo	101	17	7
Bank Marketing	1162	17	2
Churn In Telecoms	3333	21	2

EK-C(Devam)

Development Index	225	7	4
ph-recognition	653	4	15
Sloan Digital Sky Survey DR14	10000	18	3
Wine Customer Segmentation	178	14	3

EK-D: Gerçek Veriler İçin Meta Sınıflandırıcılara İlişkin Wilcoxon Test Sonuçları

Sınıflandırma Çifti	Negatif Rank	Pozitif Rank	p
AttributeSelectedClassifier-AdaBoostM1	22	59	0
Bagging-AdaBoostM1	25	63	0
ClassificationViaRegression-AdaBoostM1	21	62	0
CVParameterSelection-AdaBoostM1	86	6	0
FilteredClassifier-AdaBoostM1	24	57	0
IterativeClassifierOptimizer-AdaBoostM1	20	57	0
LogitBoost-AdaBoostM1	20	63	0
MultiClassClassifierUpdateable-AdaBoostM1	27	58	0
MultiClassClassifier-AdaBoostM1	29	64	0
MultiScheme-AdaBoostM1	86	6	0
RandomCommittee-AdaBoostM1	26	62	0
RandomSubSpace-AdaBoostM1	21	66	0
RandomizableFilteredClassifier-AdaBoostM1	54	39	0.631
Stacking-AdaBoostM1	86	6	0
Vote-AdaBoostM1	86	6	0
WeightedInstancesHandlerWrapper-AdaBoostM1	86	6	0
Bagging-AttributeSelectedClassifier	31	51	0.052
ClassificationViaRegression-AttributeSelectedClassifier	32	51	0.004
CVParameterSelection-AttributeSelectedClassifier	85	4	0
FilteredClassifier-AttributeSelectedClassifier	37	27	0.249
IterativeClassifierOptimizer-AttributeSelectedClassifier	38	44	0.118
LogitBoost-AttributeSelectedClassifier	37	47	0.051
MultiClassClassifierUpdateable-AttributeSelectedClassifier	47	38	0.073
MultiClassClassifier-AttributeSelectedClassifier	43	45	0.93
MultiScheme-AttributeSelectedClassifier	85	4	0
RandomCommittee-AttributeSelectedClassifier	32	52	0.035
RandomSubSpace-AttributeSelectedClassifier	35	44	0.158
RandomizableFilteredClassifier-AttributeSelectedClassifier	72	22	0
Stacking-AttributeSelectedClassifier	85	4	0
Vote-AttributeSelectedClassifier	85	4	0
WeightedInstancesHandlerWrapper-AttributeSelectedClassifier	85	4	0
ClassificationViaRegression-Bagging	42	38	0.46
CVParameterSelection-Bagging	87	9	0
FilteredClassifier-Bagging	49	34	0.083
IterativeClassifierOptimizer-Bagging	44	40	0.961
LogitBoost-Bagging	42	45	0.516
MultiClassClassifierUpdateable-Bagging	55	30	0.009
MultiClassClassifier-Bagging	52	37	0.478
MultiScheme-Bagging	87	9	0
RandomCommittee-Bagging	36	50	0.229
RandomSubSpace-Bagging	46	37	0.931
RandomizableFilteredClassifier-Bagging	76	16	0

EK-D(Devam)

Stacking-Bagging	87	9	0
Vote-Bagging	87	9	0
WeightedInstancesHandlerWrapper-Bagging	87	9	0
CVParameterSelection-ClassificationViaRegression	88	7	0
FilteredClassifier-ClassificationViaRegression	52	29	0
IterativeClassifierOptimizer-ClassificationViaRegression	42	39	0.536
LogitBoost-ClassificationViaRegression	38	45	0.884
MultiClassClassifierUpdateable-ClassificationViaRegression	57	30	0
MultiClassClassifier-ClassificationViaRegression	52	37	0.055
MultiScheme-ClassificationViaRegression	88	7	0
RandomCommittee-ClassificationViaRegression	42	45	0.688
RandomSubSpace-ClassificationViaRegression	47	37	0.431
RandomizableFilteredClassifier-ClassificationViaRegression	74	18	0
Stacking-ClassificationViaRegression	88	7	0
Vote-ClassificationViaRegression	88	7	0
WeightedInstancesHandlerWrapper-ClassificationViaRegression	88	7	0
FilteredClassifier-CVParameterSelection	3	84	0
IterativeClassifierOptimizer-CVParameterSelection	6	88	0
LogitBoost-CVParameterSelection	6	90	0
MultiClassClassifierUpdateable-CVParameterSelection	5	84	0
MultiClassClassifier-CVParameterSelection	9	90	0
MultiScheme-CVParameterSelection	0	0	1
RandomCommittee-CVParameterSelection	9	88	0
RandomSubSpace-CVParameterSelection	4	86	0
RandomizableFilteredClassifier-CVParameterSelection	20	76	0
Stacking-CVParameterSelection	0	0	1
Vote-CVParameterSelection	0	0	1
WeightedInstancesHandlerWrapper-CVParameterSelection	0	0	1
IterativeClassifierOptimizer-FilteredClassifier	33	50	0.031
LogitBoost-FilteredClassifier	31	54	0.008
MultiClassClassifierUpdateable-FilteredClassifier	45	32	0.72
MultiClassClassifier-FilteredClassifier	48	41	0.82
MultiScheme-FilteredClassifier	84	3	0
RandomCommittee-FilteredClassifier	34	54	0.011
RandomSubSpace-FilteredClassifier	37	44	0.051
RandomizableFilteredClassifier-FilteredClassifier	68	22	0
Stacking-FilteredClassifier	84	3	0
Vote-FilteredClassifier	84	3	0
WeightedInstancesHandlerWrapper-FilteredClassifier	84	3	0
LogitBoost-IterativeClassifierOptimizer	19	26	0.245
MultiClassClassifierUpdateable-IterativeClassifierOptimizer	57	26	0
MultiClassClassifier-IterativeClassifierOptimizer	53	33	0.018
MultiScheme-IterativeClassifierOptimizer	88	6	0
RandomCommittee-IterativeClassifierOptimizer	41	47	0.673
RandomSubSpace-IterativeClassifierOptimizer	45	41	0.625

EK-D (Devam)

RandomizableIterativeClassifierOptimizer-IterativeClassifierOptimizer	76	20	0
Stacking-IterativeClassifierOptimizer	88	6	0
Vote-IterativeClassifierOptimizer	88	6	0
WeightedInstancesHandlerWrapper-IterativeClassifierOptimizer	88	6	0
MultiClassClassifierUpdateable-LogitBoost	62	25	0
MultiClassClassifier-LogitBoost	53	32	0.003
MultiScheme-LogitBoost	90	6	0
RandomCommittee-LogitBoost	41	43	0.755
RandomSubSpace-LogitBoost	45	38	0.372
RandomizableFilteredClassifier-LogitBoost	78	16	0
Stacking-LogitBoost	90	6	0
Vote-LogitBoost	90	6	0
WeightedInstancesHandlerWrapper-LogitBoost	90	6	0
MultiClassClassifier- MultiClassClassifierUpdateable	34	49	0.019
MultiScheme-MultiClassClassifierUpdateable	84	5	0
RandomCommittee-MultiClassClassifierUpdateable	34	54	0.002
RandomSubSpace-MultiClassClassifierUpdateable	31	57	0.013
RandomizableFilteredClassifier-MultiClassClassifierUpdateable	61	32	0.002
Stacking-MultiClassClassifierUpdateable	84	5	0
Vote-MultiClassClassifierUpdateable	84	5	0
WeightedInstancesHandlerWrapper-MultiClassClassifierUpdateable	84	5	0
MultiScheme-MultiClassClassifier	90	6	0
RandomCommittee-MultiClassClassifier	34	54	0.066
RandomSubSpace-MultiClassClassifier	38	50	0.184
RandomizableFilteredClassifier-MultiClassClassifier	72	22	0
Stacking-MultiClassClassifier	90	6	0
Vote-MultiClassClassifier	90	6	0
WeightedInstancesHandlerWrapper-MultiClassClassifier	90	6	0
RandomCommittee- MultiScheme	9	88	0
RandomSubSpace- MultiScheme	4	86	0
RandomizableFilteredClassifier- MultiScheme	20	76	0
Stacking- MultiScheme	0	0	1
Vote- MultiScheme	0	0	1
WeightedInstancesHandlerWrapper- MultiScheme	0	0	1
RandomSubSpace- RandomCommittee	53	35	0.16
RandomizableFilteredClassifier- RandomCommittee	80	14	0
Stacking- RandomCommittee	88	9	0
Vote- RandomCommittee	88	9	0
WeightedInstancesHandlerWrapper- RandomCommittee	88	9	0
RandomizableFilteredClassifier- RandomSubSpace	75	16	0
Stacking- RandomSubSpace	86	4	0
Vote- RandomSubSpace	86	4	0
WeightedInstancesHandlerWrapper- RandomSubSpace	86	4	0
Stacking- RandomizableFilteredClassifie	76	20	0
Vote- RandomizableFilteredClassifier	76	20	

EK-D(Devam

WeightedInstancesHandlerWrapper-RandomizableFilteredClassifier	76	20	0
Vote- Stacking	0	0	1
WeightedInstancesHandlerWrapper- Stacking	0	0	1
WeightedInstancesHandlerWrapper- Vote	0	0	1

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı	:	R*****n A**Z
Doğum Yeri ve Tarihi	:	D*****i 0*0*1**9

Eğitim Durumu

Lisans Öğrenimi	:	Muğla Sıtkı Koçman Üniversitesi İstatistik Bölümü
Bildiği Yabancı Diller	:	İngilizce
Bilimsel Faaliyetler	:	11. Fen Bilimleri Araştırma Sempozyumu, Muğla Sıtkı Koçman Üniversitesi, Ocak 2021, Katılımcı

İletişim

Adres	:	İ******şa Ma**llesi 3*9. Sokak */2 F**a/İ***R
Telefon	:	+90 5*5 4*4 8* 8*
E-Posta Adresi	:	r*********z@gmail.com