T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SÜREKLİ VERİ SETLERİNİN FARKLI AYRIKLAŞTIRMA YÖNTEMLERİ KULLANILARAK KURAL TABANLI SINIFLANDIRMA ALGORİTMALARINA UYARLANMASI VE KARŞILAŞTIRILMASI

Sanarya Samal FAROOQ FAROOQ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Nisan-2019
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS

Sürekli Veri Setlerinin Farklı Ayrıklaştırma Yöntemleri Kullanılarak Kural Tabanlı Sınıflandırma Algoritmalarına Uyarlanması ve Karşılaştırılması

Sanarya Samal FAROOQ FAROOQ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr.Öğr.Üyesi Murat SELEK

2019, 40 Sayfa

Jüri
Dr.Öğr.Üyesi Murat SELEK
Prof.Dr. Fatih BAŞÇİFTÇİ
Doç.Dr. Halife KODAZ

Veri ayrıklaştırma, veri madenciliğinde kullanılan en popüler veri ön işleme tekniklerinden biridir. Veri ayrıklaştırma işleminde sürekli olan değerler belirli aralıklara göre ayrık değerlere dönüşür. Literatürde birden fazla ayrıklaştırma yöntemi bulunabilir, fakat bu ayrıklaştırma yöntemlerinin içinde en yaygın olanlar Eşit Genişlik (EG), Eşit Frekans (EF) ve Entropi yöntemleridir. Her ayrıklaştırma yönteminin kendine göre avantaj ve dezavantajları bulunmaktadır. Veri ayrıklaştırma işlemi veri kümeleri üzerinde birçok işlemi kolaylaştırır ve hızlandırır.

Literatürde araştırmacılar tarafından geliştirilen birçok ayrıklaştırma teknikleri bulunmaktadır. Geliştirilen bu ayrıklaştırma yöntemleri veri madenciliğinde sınıflandırma başarısını artırması, verilerin hızlı ve basit bir şekilde işlenmesi gibi birçok amaç için kullanılmaktadır. Veri madenciliğinde kural tabanlı sınıflandırma algoritmaları ayrık değerli özeliklere sahip olan veri kümelerinde daha iyi performans ve sınıflandırma başarıları göstermektedir. Bu çalışmanın amacı sürekli değerli veri özeliklerine sahip olan veri kümelerinin özellik değerlerini EG, EF ve Entropi Tabanlı ayrıklaştırma yöntemleri ile ayrıklaştırarak, Karar Ağacı ID3, Naive Bayes (NB) ve Karar Tablo (KT) Kural Tabanlı sınıflandırma algoritmalarına uyarlanması ile daha iyi bir sınıflandırma başarısı elde etmektir. Bu çalışmada, UCI veri seti içinden sıkça kullanılan 13 adet sürekli değerli özelliklere sahip olan veri seti alınmaktadır. Bu veri setlerinin özellik değerleri ayrık değerlere dönüştürülerek farklı sınıflandırma algoritmaları üzerine uyarlanmaktadır. Bu sayede sınıflandırma algoritmasına verilen ayrıklaştırılmış verilerin sınıflandırma başarısının artması sağlanacaktır.

Anahtar Kelimeler: Ayrıklaştırma işlemi, Sınıflandırma, Eşit genişlik, Eşit frekans, Entropi

ABSTRACT

MS THESIS

Adaptation and Comparison of Continuous Data Sets to Rule-Based Classification Algorithms by Using Different Discretization Methods

Sanarya Samal FAROOQ FAROOQ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE OF PHILOSOPHY
IN COMPUTER ENGINEERING

Advisor: Assist.Prof.Dr. Murat SELEK

2019, 40 Pages

Jury
Assist.Prof.Dr. Murat SELEK
Prof.Dr. Fatih BAŞÇİFTÇİ
Assoc.Prof.Dr. Halife KODAZ

Data discretization is one of the most popular data preprocessing techniques used in data mining. Values that are continuous in the data discretization process become discrete values over certain intervals. In literature, more than one discretization method can be found but the most common methods are equal width, equal frequency, and entropy. Each discretization method has its own advantages and disadvantages. Data discretization process makes easier and faster many operations on data sets. In literature there are many discretization techniques that developed by researchers. These discretization methods are used for many purposes such as increasing the classification success in the data mining, processing the data quickly and simply. Rule-based classification algorithms in data mining show better performance and classification success in data sets with discrete properties. The aim of this thesis is to obtain a better classification performance by adapting the property values of data sets with continuous valuable data properties by equal width, equal frequency and entropy based discretization methods and adaptation to decision tree ID3, Naive Bayes and decision table rule based classification algorithms. The importance of this thesis is to increase the classification performance on rule-based classification algorithms of continuous data sets. In this thesis, a data set which has 10 continuously valuable properties which are frequently used from UCI data sets will be taken. The property values of these data sets are transformed into discrete values and are adapted to rule-based classification algorithms. Thus the classification success of the discrete data which given to the classification algorithm will be increased.

Keywords: Discretization, Classification, Equal width, Equal frequency, Entropy

ÖNSÖZ

Öncelikle, bana bu yüksek lisans yolculuğumda yardımcı olan herkese teşekkür ediyorum. İlk olarak bana hem maddi hem de manevi destek verdikleri için aileme annem ve babama çok teşekkür ederim, daha sonrası bana yardım eden ve destek veren danışmanım Dr.Öğr.Üyesi Murat SELEK'e ve yakın arkadaşlarıma çok teşekkür ediyorum. Bana yüksek lisans esnasındaki hastalığımda yardımcı olanlara çok teşekkürler, hepiniz iyi ki varsınız.

Sanarya Samal FAROOQ FAROOQ
KONYA-2019

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

İÇİNDEKİLER ..... **vii**

KISALTMALAR ..... **ix**

ŞEKİLER VE ÇİZELGELER ..... **x**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **5**

3. MATERYAL VE YÖNTEM **14**

3.1. Veri Setleri 14
3.2. Yöntemler 15
3.2.1. Sınıflandırma işlemi 15
3.2.2. Ayrıklaştırma işlemi 19

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA **27**

5. SONUÇLAR VE ÖNERİLER **37**

5.1 Sonuçlar 37
5.2 Öneriler 38

KAYNAKLAR **39**

EKLER **42**

ÖZGEÇMİŞ **46**

KISALTMALAR

Kısaltmalar

VM	:	Veri Madenciliği
ID3	:	Geliştirilmiş Karar Ağacı Algoritması (Improved Decision Tree Algorithm)
MDLP	:	Sürekli Değerli Özelliklerin Çok Aralıklı Ayrıklaştırılması (Multi-Interval Discretization of Continuous-Valued Attributes)
1R	:	1-kural (1−rule)
NP-ZOR	:	Belirsiz Polinom (Nondeterministic Polynomial)
NCAIC	:	Yeni Sınıf Özelliği Bağımlılığı (Novel Class-Attribute Interdependency)
SMDNS	:	Denetimli ve Çok Değişkenli Ayrıklaştırma (Supervised and Multivariate Discretization)
NB	:	Saf Bayes Algoritması (Naive Bayes Algorithm)
BNs	:	Bayesian Ağları (Bayesian Networks)
DTs	:	Karar ağaçları (Decision Trees)
RLs	:	Kural Öğrenciler (Rule-Learners)
OHS	:	Yumurtalık hiper stimülasyonu (ovariel hyper stimulation)
RST	:	Kaba Kümeler Teorisi (Rough Sets Theory)
HWDM	:	Haar dalgacık ayrıklaştırma yöntemi (Haar wavelet discretization method)
FDM	:	Sonlu farklar yöntemi (Finite difference method)

ŞEKİLER VE ÇİZELGELER

ŞEKİLLER

Şekil 1.1 Ayrıklaştırma yöntemlerinin arasındaki ilişkiler
Şekil.4.1 İlk ekran görüntüsü (visual studio).
Şekil.4.2 Uyguladığmızdaki ekran görüntüsü (visual studio).
Şekil.4.3 weka programının ilk ekran görüntüsü.
Şekil.4.4 weka prgramının sınıflandırma sonucunun ekran görüntüsü

ÇİZELGELER

Çizelge3. 1 Medikal veri tabanı eğitim seti 15
Çizelge3. 2 Medikal veri tabanı tahmin seti 16
Çizelge 3.3 Eşit Genişlik Ayrıklaştırma yöntemi için örnek 23
Çizelge3 .4 Eşit Frekans Ayrıklaştırma yöntemi örneği 24
Çizelge 3 .5 K-ortalama yöntemine örnek 24
Çizelge 3.6 Entropi Tabanlı Ayrıklaştırma yöntemi örneği 26
Çizelge 4.1 EG aralık 3, 5 ve 7 için sınıflandırma başarıları 27
Çizelge 4.2 EG aralık 9, 11 ve 13 için sınıflandırma başarıları 27
Çizelge4 .3 EG aralık 15, 17 ve 19 için sınıflandırma başarıları 28
Çizelge 4.4 EG aralık 21, 23 ve 25 için sınıflandırma başarıları 28
Çizelge 4.5 EF Aralık 3, 5 ve 7 için sınıflandırma başarıları 30
Çizelge 4.6 EF Aralık 9, 11 ve 13 için sınıflandırma başarıları 30
Çizelge 4.7 EF Aralık 15, 17 ve 19 için sınıflandırma başarıları 30
Çizelge 4.8 EF Aralık 21, 23 ve 25 için sınıflandırma başarıları 31
Çizelge4 .9 EG, EF ve ID3 ayrıklaştırma yöntemlerinin sınıflandırma başarı sonuçları 33
Çizelge 4.10 EW, EF, ID3 ve diğer ayrıklaştırma yöntemlerinin NB sınıflandırma başarının sonuçları 34
Çizelge 4.11 EW, EF, ID3 ve diğer ayrıklaştırma yöntemlerinin ID3 sınıflandırma başarının sonuçları 34

1. GİRİŞ

Veri Madenciliği (VM), Gerçek bir veriye bir VM algoritmasının başarılı bir şekilde uygulanmasını garantilemek için, sorunsal anlama, veri anlama veya veri ön işleme gibi bazı görevler gelirilmelidi. Veri önişleme, VM alanında çok önemli bir araştırma konusudur ve veri dönüşümü, temizleme ve veri azaltma gibi çeşitli işlemleri içerir (Witten ve ark., 2016a).

Temel veri azaltma tekniklerinden biri olan ayrıklaştırma,son yıllarda giderek artan bir ilgiye sahip olmuş ve VM'de en yaygın olarak kullanılan önişleme tekniklerinden biri haline gelmiştir (Cios ve ark., 2007). Ayrıklaştırma işlemi, nicel veriyi nitel veriye yani sayısal özniteliklere, sonlu sayıda aralıklarla ayrı veya nominal özniteliklere dönüştürerek, sürekli bir alanın birbirine geçmeyen bir bölümünü elde eder (Dougherty ve ark., 1995a). Ardından her aralık ile sayısal ayrık bir değer arasında bir ilişki kurulur. Uygulamada ayrıklaştırma, büyük bir spektrumdaki sayısal değerlerden oluşan veriyi, harici değerlerin büyük ölçüde küçülmüş bir alt kümesine eşleştirdiğinden, veri azaltma yöntemi olarak görülebilir. Sayısal veriler üzerinde ayrıklaştırma işlemi gerçekleştirildikten sonra VM bu verileri ayrık veriler olarak işlemlere tabi tutar. Birçok mevcut VM algoritması, ayrık veriler kullanarak öğrenmelerini gerçekleştirebilir, fakat gerçek dünya uygulamaları genelde sürekli veriler içerir. Bu sayısal verilerin, bu gibi VM algoritmaları kullanmadan önce ayrık hale getirilmesi gerekmektedir. Denetimli öğrenmede ve özel olarak sınıflandırmada, ayrıklaştırmayı aşağıdaki gibi tanımlanabilir: N örnekten ve C hedef sınıflarından oluşan bir veri seti varsayarsak, bir ayrıklaştırma algoritması, bu veri kümesindeki örnekleri ayrı aralıklara ayırır. $D = \{[d_0, d_1], [d_1, d_2], [d_2, d_m]\}$ (1.1)

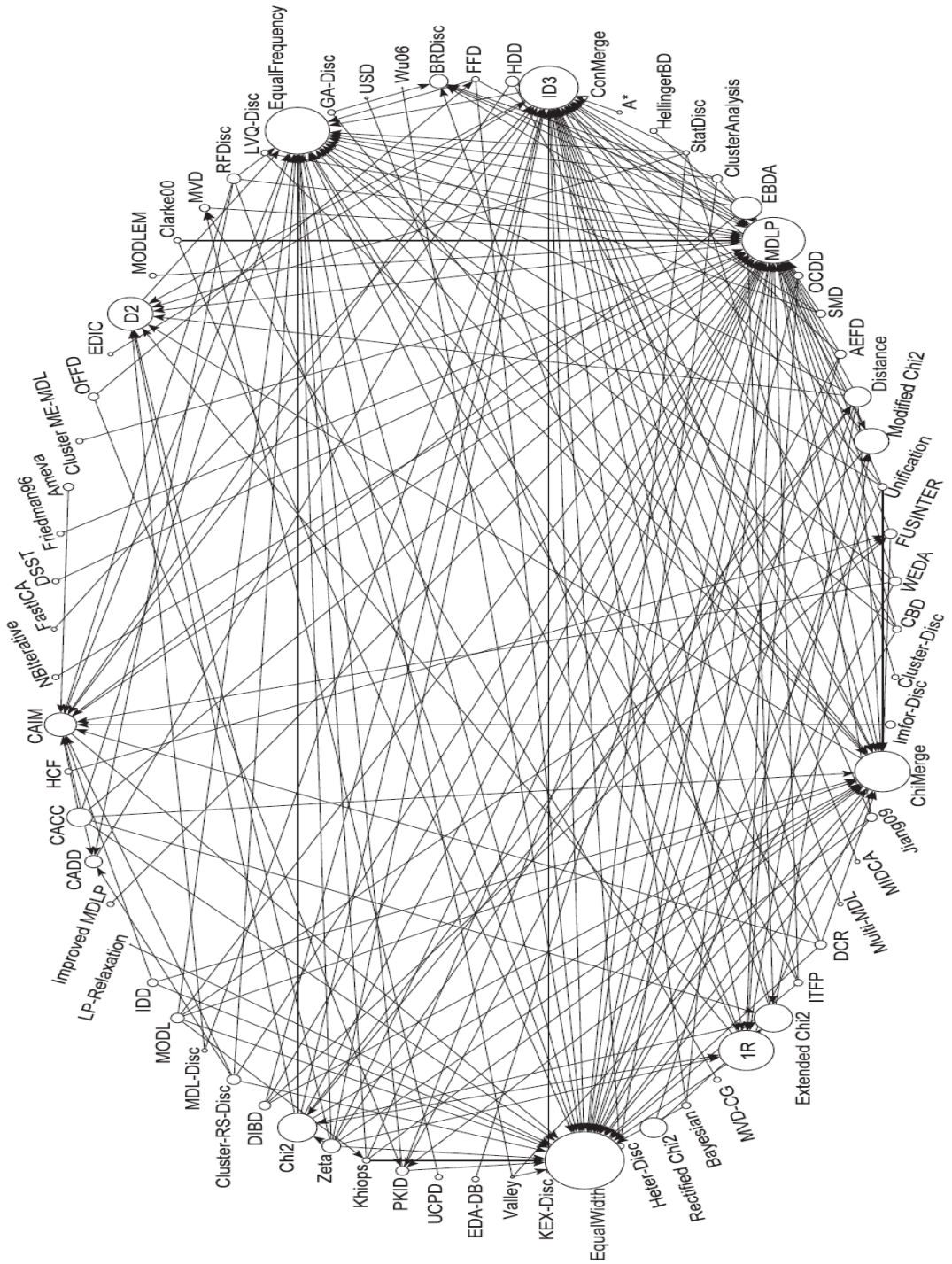
Burada, d_0 minimum değerdir, d_m maksimum değerdir ve $d_i < d_{i+1}$, i = 0, 1, ..., m-1'dir. Böyle bir ayrıklaştırma sonucunda sayısal verilerin kesme noktalarının kümesi aşağıdaki gibidir.

$$P = \{d_1, d_2, \dots, d_{m-1}\} \quad (1.2)$$

Veri üzerinde ayrıklaştırma kullanmanın gerekliliği çeşitli faktörlerden kaynaklanabilir. Birçok VM algoritması, sadece ayrık değerler üzerinde çalışabilir (Dougherty ve ark., 1995a). Örneğin, VM'de ilk 10 olarak kabul edilen 10 metottan üçü olan C4.5, Apriori ve Naive Bayes (NB) verilerin ayrık olmasına ihtiyaç duyar. Bu tür

VM algoritmalarında sürekli veriler uygulandığı zaman öğrenme daha az etkin ve etkili olabilir (Wu ve Kumar, 2009). Ayrıklaştırmadan türetilen diğer avantajlar, verilerin azaltılması ve basitleştirilmesi, öğrenmenin daha hızlı hale getirilmesi ve daha doğru, kompakt ve daha kısa sonuçlar vermesi ve veride muhtemelen gürültünün azalmasıdır. Hem araştırmacılar hem de uygulayıcılar için ayrık nitelikleri anlamak, kullanmak ve açıklamak daha kolaydır. Bununla birlikte, herhangi bir ayrıklaştırma işlemi genellikle bilginin kaybolmasına yol açar. Bu türden bilgilerin en aza indirgenmesi, bir ayrıklaştırıcının temel hedefini kaybettirir. Optimal ayrıklaştırmanın elde edilmesi NP-Zor problemidir (Chlebus ve Nguyen, 1998). Literatürde çok sayıda ayrıklaştırma tekniği bulunmaktadır. Somut bir problem veya veri seti ile uğraşırken, bir ayrıklaştırıcının seçiminde, arka öğrenme görevinin başarısını, doğruluğunu, modelin basitliğini, vb. şartları göz önüne almak gerekir. Ayrıklaştırma için farklı sezgisel yaklaşımlar önerilmiştir. Örneğin, bilgi entropisine dayalı yaklaşımlar, istatistiksel 2 testi, olasılık, kaba setler vb. tek değişkenli/çok değişkenli, denetlenen/denetlenmeyen, yukarıdan aşağıya/darboğaza, küresel/yerel, statik/dinamik ve daha fazlası gibi ayrıklaştırıcıların sınıflandırılmasını sağlamak için başka kriterler kullanılmıştır. Bütün bu ölçütler, hâlihazırda önerilen taksonomilerin temelini oluşturmaktadır. Her durum için en iyi ayrıklaştırıcının belirlenmesi ve gerçekleştirilmesi çok zor bir iştir. Ancak temsilci bir öğrenme grubu ve ayrıklaştırıcı grubu göz önüne alarak kapsamlı deneyler yapmak en iyi sonucu vermeye ve karar vermeye yardımcı olabilir (Bakar ve ark., 2009). Ayrıklaştırma tekniklerinin bazı özeliklerinin literatürde incelemeleri yapılmıştır. Örneğin, ayrıklaştırma yöntemlerinin içinde statik/dinamik, global/local, tek değişkenli / çok değişkenli ve denetimli/denetimsiz özelikler bulunabilir (Liu ve ark., 2004). Bu ayrıklaştırma yöntemlerinin özelikleri 3. bölümde incelenmiştir.

Literatürde bulunan tüm ayrıklaştırma yöntemleri Şekil.1.1'de detaylı bir şekilde gösterilmiştir. Şekil.1.1'de her düğüm bir ayrıklaştırma yöntemini temsil etmektedir, oklar ise ayrıklaştırma yöntemlerinin arasındaki benzerlik ilişkisini göstermektedir. Literatürde en çok karşılaştırılan ayrıklaştırma yöntemleri Eşit Genişlik (EG), Eşit Frekans (EF), MDLP, ID3, ChiMerge, 1R, D2 ve Chi2'dir (Garcia ve ark., 2013).



Şekil 1.1 Ayrıklaştırma yöntemlerinin arasındaki ilişkiler (Garcia ve ark., 2013)

Bu benzerliklerden dolayı, çalışma üç temel hedefe ayrılabilir:

- Ayrıklaştırma yöntemlerinde gözlenen temel özelliklere dayalı olarak tam bir taksonomi önermek. Taksonomi, teorik açıdan bir ayrıklaştırıcı seçmek için avantajları ve dezavantajları belirlenmesi sağlayacaktır.

- En temsili ve en yeni ayrıklaştırıcıları, elde edilen aralıkların sayısı ve tutarsızlık düzeyi açısından analiz eden deneye dayalı bir çalışma yapmak.
- Son olarak, bir dizi temsilci VM modeli için en iyi ayrıklaştırıcıları, tahmin sınıflandırma başarısını ölçmek için iki metrik kullanarak ilişkilendirmek.

Deneysel çalışmada, parametrik olmayan testlere dayalı istatistiksel bir analiz yer almaktadır. Toplam 3 ayırklaştırıcıyı içeren deneyler yapılarak; karar ağaçlarına ve Bayes öğrenme ailelerine ait altı sınıflandırma yöntemi ve 4 veri seti kullanılmıştır. Deneysel değerlendirme, eldeki verileri göz önünde bulundurarak, her ayrıklaştırıcı için en iyi parametrelerin kapsamlı bir araştırmasına karşılık gelmektedir. Ardından, ana odak noktası, en iyi performans gösteren ayrıklaştırıcıların bir alt kümesini her klasik sınıflandırıcıya genel bir yapılandırma kullanarak düzgün bir şekilde ilişkilendirmektir.

2. KAYNAK ARAŞTIRMASI

Makine Öğrenmesi ve VM uygulamalarının çoğu yalnızca ayrık özellik değerlerine uygulanabilir. Bununla birlikte, gerçek dünyadaki veriler genellikle süreklidir (Han ve ark., 2006).

Veri kümesindeki özelliklerin değerleri sürekli olduğunda bu veri madenciliği algoritmalarının başarısını etkiler (Hacibeyoglu ve ark., 2011).

Bu nedenle, ayrıklaştırma, temsil etmek ve belirtmek için daha özlü olan sayı aralıklarını bulmak suretiyle bu sorunu giderir. Sürekli özniteliklerin ayrık hale getirilmesi, bilgi çıkarma sürecindeki önemli veri ön işlem adımlarından biridir. Etkili bir ayrıklaştırma yöntemi, VM ve makine öğrenme algoritmasının verimliliğini artırmakla kalmaz aynı zamanda ayrık veri setinden elde edilen bilgiyi daha kompakt, anlaşılması kolay ve kullanışlı hale getirir (Dash ve Paramguru, 2011).

Genelde ayrıklaştırma yöntemleri Denetimli ve Denetimsiz olarak ikiye ayrılır ve bu ayrıklaştırma yöntemlerinin avantaj ve dezavantajları bulunmaktadır (Peng ve ark., 2009).

Yapılan araştırmaya göre denetimli ayrıklaştırma yöntemleri denetimsiz ayrıklaştırma yöntemlerinden VM'nin sınıflandırma problemlerinde daha iyi sonuç vermektedir (Hacıbeyoğlu ve Ibrahim, 2016).

Denetimsiz ayrıklaştırma yöntemlerinin en büyük dezavantajı ise interval dediğimiz parça sayılarının kullanıcı tarafından verilmesidir. Dolayısıyla optimum parça sayılarının tahmini NP-Zor bir problemdir (Garcia ve ark., 2013).

Araştırmacı Marc Boulle tarafından denetimsiz olan EF ayrıklaştırma yöntemi için veri seti üzerinden bulunan parça sayısı yöntemi önerilmiştir (Boulle, 2005).

Araştırmacılar tarafından bu güne kadar çok sayıda ayrıklaştırma yöntemi geliştirilmiştir. Örneğin, Ankit Gupta ve arkadaşları tarafından 2010 yılında kümeleme tabanlı yeni bir ayrıklaştırma yöntemi önerilmiştir. Önerilen ayrıklaştırma yönteminin performansını UCI veri kümeleri üzerinde test etmişlerdir. Deneysel sonuçlarına göre geliştirilen kümeleme algoritması özelikle sınıflandırma algoritmaları için çok iyi bir performans sergilemiştir (Gupta ve ark., 2010a).

Deqin Yan ve arkadaşları tarafından 2013 yılında sınıf bilgisini kullanan yeni bir sınıf-nitelik bağımlılığı kriteri (a Novel Class-Attribute Interdependency Criterion-NCAIC) adında yeni bir ayrıklaştırma yöntemi geliştirilmiştir. Geliştirilen denetimli

ayrıklaştırma yöntemini sınıflandırma problemleri üzerinde uygulamışlardır. Geliştirilen ayrıklaştırma yöntemini, UCI veri tabanından alınan 13 veri kümesi ve çok bilinen C4.5 karar ağacı sınıflandırma algoritması üzerinde test etmişlerdir. Sonuçları, önerilen ayrıklaştırma yönteminin sınıflandırma problemleri için çok uygun ve olumlu olduğunu göstermiştir (Yan ve ark., 2014b).

Feng Jiang ve arkadaşı tarafından 2015 yılında denetimli bir ayrıklaştırma yöntemi denetimli ve çok değişkenli bir ayrıklaştırma algoritması (a Supervised and Multivariate Discretization Naive Scaler-SMDNS) isimli yöntem önermişlerdir. Önerilen ayrıklaştırma yöntemi kaba küme (rough sets) mantığı ile çalışmaktadır. Önerilen ayrıklaştırma yöntemi karar ağaçlı ID3 sınıflandırma algoritması üzerinde çok iyi sınıflandırma başarısı sergilemiştir (Jiang ve Sui, 2015).

Madhu ve arkadaşı biyomedikal verileri üzerinde çok iyi bir performans gösteren z-score adında yeni bir ayrıklaştırma yöntemi geliştirmişlerdir. Deneysel sonuçları, sınıflandırma algoritmasının verimliliğini artırdığını ve aynı zamanda karar verme süreci için sınıflandırıcı karışıklığını en aza indirgediğini göstermektedir (Madhu ve ark., 2014).

Her ayrıklaştırma yönteminin temel avantajlarını bilmek önemlidir. Bugüne kadar geliştirilen birçok ayrıklaştırıcı, araştırmacılar tarafından deneysel olarak analiz edilmiştir. Ancak en iyi performans göstereninin hangisi olduğu konusunda tek bir sonuç yoktur (Garcia ve ark., 2013).

Denetimli makine öğrenmesinde, bazı algoritmalar ayrık veri ile sınırlıdır ve sürekli özniteliklerin ayrılması gerekir. Ayrıklaştırma yöntemleri için istatistiksel kriterlere göre bilgi içeriği ve diğer özel kriterler incelenmiştir. Bu çalışmada ayrıklaştırma yöntemi Khiops, önerilmiş 1 tabanlı chi-square istatistiğine göre incelenmiştir. Makine ile öğrenme algoritmaları, kesikleştirme özelliği gerektirir. Bu çalışmada sürekli özelliklerin kesikleştirilmesi ile bu konudaki bazı yöntemlerin deneysel değerlendirmeleri hakkındaki önceki çalışma gözden geçirilmiştir. Ayrıca, bir güdümsüz kesikleştirme yöntemini güdümlü algoritmalardan olan entropi tabanlı ve saflık tabanlı yöntemlerle karşılaştırılmıştı. Entropi temelli bir yöntem kullanarak özellikleri kesikleştirildiğinde Naive-Bayes algoritma performansının önemli ölçüde arttığını bulunmuştu. Aslında teste tabi tutulan veri kümeleri üzerinde uygulanan kesikleştirme yönteminin Naive Bayes versiyonu C4.5 ortalamasını az da olsa geçmektedir. Bazı durumlarda, özelliklerin önceden kesikleştirilmesi halinde C4.5 başlatma algoritması performansının önemli ölçüde arttığını gösterilmiştir. Yapılan

deneylerde, yerel kesikleştirme yeteneği özelliklerine sahip C4.5'i dikkate aldındığında performans hiçbir zaman kayda değer bir düşme göstermemiştir (Dougherty ve ark., 1995b).

Denetlenen makinedeki bazı algoritmalar sadece kesikleştirilmiş veriye özgü kılınmış olup süreklilik arzeden öznitelikleri kesikleştirme özelliği de bulunmuştur. İstatistiksel kriterler üzerine kurulu kesikleştirme yöntemleri ile bilgi içeriği üzerinde geçmiş zamanda çalışılmıştır. Bu çalışmada ise ki-kare istatistikleri üzerine kurulu Khiops-1 kesikleştirme yöntemi öneri olarak sunulmuştur. İlgili Ki-Birleştir (ChiMerge) ve Ki-Ayır(ChiSplit) yöntemlerine zıt olarak bu yöntem tüm kesikleştirme alanındaki ki-kare kriterlerini küresel bir biçimde optimize etmekte ve durdurma kriterlerine ihtiyaç duyulmamıştır. Bir teorik çalışma ve ardından deneyler vasıtasıyla yöntemin sağlamlığı ve kestirim gücü gösterilmiştir (Boulle, 2004).

Gerçek veri karmaşık bir biçimde, kesik ve sürekli olarak gelmekte ve başlatmada da kesik veriye gerek duyulmuştur. Süreklilik arzeden özniteliklerin etkili bir biçimde kesikleştirilmesi hız, hassasiyet ve anlaşılırlık bakımından önemli bir problem teşkil etmiştir. Bu çalışmada Bayesian yaklaşımı üzerine kurulu yeni bir kesikleştirme yöntemi olan MODL1'i önerilmiştir. Bu model alan üzerine tanımlanmış bir kesikleştirme modelleri uzayı ve bu uzaydaki öncel dağılım yapılmasını göstermiştir. Bu da kesikleştirme konusunda Bayes optimal değerlendirme tanımını ortaya çıkarmıştır. O nedenle optimuma en yakın kesikleştirmeleri bulunmasını sağlayacak yeni bir super lineer optimizasyon algoritması önermiştir. Deneyler, bu yeni kesikleştirme yönteminden elde edilen yüksek başlama performanslarına işaret etmiştir (Boullé, 2006).

Hem sürekli hem de kesikleştirilmiş değişkenler içeren verilerden gelen BN öğrenimi açısından kesikleştirme problemine değinilmiştir. Çok değişkenli kesikleştirilmeler için diğer değişkenlerle ilişkilerin dikkate alındığı yeni bir teknik sunulmuştur. Bu teknik, verilen BN yapısı ve gözlemlenen veri üzerinde sürekli değişen bir değişken için uygulanan kesikleştirme politikasının bir Bayesian yöntemi ile ölçeklendirilmesine dayanmaktadır. Ölçeklendirme üzerinde işlem yapılan mevcut BN yapısı ile ilişkili olduğundan, bir değişkenin kesikleştirilmesi de BN yapı değişkenlerine göre dinamik bir biçimde kesikleştirilmesi ihtiyacına dayanmaktadır (Monti ve Cooper, 1998).

Hata temelli veya entropi temelli yöntemlerin eşliğinde bu çalışma hem karşılaştırma hem de senaryoların analizini içermektedir ki, burada hata enazlama

uygun bir kesikleştirme kriteri değildir. C4.5 karar ağacı algoritması üzerine kurulu yeni bir kesikleştirme yöntemi sunmakta ve bu yöntemi mevcut bir Minimum Tanım Uzunlu ğu Prensibi ile son önerilen hata tabanlı teknik kullanan entropi temelli algoritma ile karşılaştırılmıştır. Bu kesikleştirme yöntemlerini C4.5 ve Naive-Bayesian tasnifleri altında UCI tarafından sağlanan bilgi havuzundan elde edilerek veri setleri ile değerlendirilmiş ve hesaplama karmaşıklığının analizi yapılmıştır. Elde edilen sonuçlar, entropi temelli MDL sezgisel sonuçlarının ortalamada hata enazlamadan daha iyi sonuç verdiğini göstermektedir. Ardından hata temelli yaklaşımların entropi temelli yöntemlerle karşılaştırmasını yaparak ortaya çıkan kusurlarının analizini yapılmıştır (Kohavi ve Sahami, 1996).

Burada makine öğrenme algoritmalarının kesikleştirilmesi problemine değinilmiştir. Mevcut süreçler, bu amaca yönelik değişkenler arasındaki bağımlılığı dikkate alınmamıştır. Önerilen yöntem bu türden bir bağımlılığı açığa çıkaran kümelemeler kullanmıştır. Sonuçlar, bu sürecin her durumda tasnif performansını arttırdığını ve değişkenlerin ilk olarak kesikleştirilmesi halinde daha iyi bir performans elde edildiğini göstermiştir. Bu türden bir kesikleştirme aynı zamanda genel olarak hesaplama süresini de azalmıştır (Gupta ve ark., 2010b).

Gerçek değerli öznitelikler arzeden problemleri çözmek için Karar Ağacı (DTs), BN ve Karar-Öğreniciler (RLs) gibi yöntemler kullanılarak yapılmış bir kesikleştirme algoritmasında sonuç aşamalarına nominal değerler işlemi uygulanmıştır. Bu sistemler, doğru aşamaların seçilmesi ile ilişkilidir. Bir algoritmanın bu türden bir işleme özel bilgi kaybı ile makul kesim noktaları bulma arasındaki dengeyi kurması gerekmiştir. Bu çalışma iyi bilinen kesikleştirme tekniklerini sunmuştur ve verilen referansların önemli teorik konuları kapsadığını ayrıca araştırmacılara ilginç araştırma yolları sağlayıp muhtemel birleşimler konusunda tavsiyelerde bulunmuştur (Kotsiantis ve Kanellopoulos, 2006a).

Kesikleştirme, pek çok bilgi keşfi ile veri madenciliği işlerinde kullanılmaktadır Kesikleştirme ile süreklilik arzeden bir takım öznitelikler aralara kategorik değerler atayarak kesik kesik hale getirilmiştir ve böylece nicel veri nitel veriye dönüştürülmüştür. Sembolik veri madenciliği algoritmaları sürekli veri üzerine uygulanmak suretiyle bu tür akan very daha özlü ve daha belirgin hale getirilmiştir. Literatürde çok sayıda kesikleştirme önerileri bulunmaktadır üzere ve bütün bu öneriler için sınıflandırma teşebbüsleri de bulunmakradır üzere bununla birlikte önceki çalışmalarda özelliklerin tanımlanmasında bir mutabakat eksikliği bulunmakta ve henüz

hiçbir resmi tasnif yoktur. Sadece çok az sayıda kesikleştirme yapanlar genel olarak dikkate alınmıştır. Bu çalışmada ise hem teorik hem de empirik açılardan literatürde yer alan kesikleştirme yöntemlerinin bir araştırması sunulmuştur. Önceki çalışmada işaret edilen ana özellikleri temel alan bir sınıflandırma geliştirlmiştir ve böylece bugüne kadarki bütün bilinen yöntemler dâhil olmak üzere simgeleme sistemlerini birleştirmişler. Güdümlü sınıflandırmada deneysel bir çalışma yürütülmüş ve konuyu en iyi temsil eden en yeni kesikleştiricileri, farklı sınıflandırmaları ve hassasiyetle ölçülmüş çok sayıda veri setini, en iyi temsil eden en yeni kesikleştiricilerle karşılaştırmışlar. Buradaki kararsızlık, parametrik olmayan istatistikî testler aracılığıyla doğrulanmış ve veri madenciliği amacıyla süreklilik arz eden öznitelikler kullanılarak kesikleştirilmiştir(Garcia ve ark., 2013).

Çoğu entropi tabanlı kesikleştirme yöntemi yereldir ve kıymetli veriler kolaylıkla kaybedilebilir. Burada yeni bir entropi tabanlı algoritma sunmak üzeri. Kesikleştirme planının başlangıcında kesme noktaları belirlenmiştir. Böylelikle yöntemi oluşturan kesikleştirme sürecindeki etkileşim global bir özellik kazanmıştır. Deney de göstermektedir ki, aynı C4.5 kural üreteci ile yöntemi mevcut entropi tabanlı kesikleştirme yöntemlerinden daha güçlü kurallar üretebilmiştir (Li ve Wang, 2002).

Yan ve arkadaşları 2014te kesikleştirme, süreklilik arz eden öznitelikleri öğrenme sistemlerinde ileri aşamalarda daha fazla veri işleme için sinyal sunma amacıyla kesik formata dönüştürme sürecidir. Kesikleştirme tekniklerindeki asıl sorun, süreklilik arz eden değerlerin bilgi kaybını en aza indiren bir ortamı ve veriyi en verimli şekilde karakterize eden sınırlı sayıda aralar kullanarak en iyi sunum yolunu bulmaktır.şimdiye kadar karşılaşılmamış türden veri dağıtımını ve her türden sınıflar ve öznitelikler arasındaki bağımlılığı dikkate alan bir özellik bağlarını kesikleştirme yöntemi (NCAIC) sunlmuştur. Çözümleri veri setlerine kabaca yaklaşım biçimi uygulanan kesikleştirme algoritmalarının ana gövdesini oluşturmakta ve sınıf-öznitelik karşılıklı bilgisi otomatik kontrol için ve sürekli özniteliklerin kesikleştirilmesi sürecini düzenleme amacıyla kullanılmaktadır. Diğer beş ayrı kesikleştirme algoritması ile gerçekleştirilmiş bazı deneylerin karşılaştırmalı sonuçları verilmiştir ki, orada UCI veri tabanından 13 kıstas veri seti ayrıştırılmış ve bunun için iyi bilinen C4.5 karar ağacı kullanılmıştır. Sonuçlar, genel olarak önerdikleri algoritmanın sınıflandırma açısından diğer test edilen kesikleştirme algoritmalarından daha iyi sonuç verdiğini gösterilmiştir (Yan ve ark., 2014a).

Veri madenciliği ve makine öğrenimine ilişkin çoğu algoritma sadece ve sadece kesikleştirilmiş öznitelikler ile işlem yapabilmektedir. Bu algoritmaları kullanabilmek için özniteliklerden bazılarının sayısal olması durumunda sayısal özniteliklerin kesikleştirilmesi gerekmektedir. Normal dağılımın genel olması sebebiyle, normal dağılım üzerini kurulu jiang ve arkadaşları tarafınan yaklaşık bir eşit frekans kesikleştirme yöntemi sunulmuştur. Yöntemin uygulanması basittir. Bu yöntemin hesaplamadaki karmaşıklığı neredeyse veri setinin boyutuyla aynı orandadır ve çok büyük veri setleri için de kullanılabilir. Yöntemi UCI veri setleri üzerinde diğer bazı kesikleştirme yöntemleri ile karşılaştırmışlar. Deney sonuçları bu güdümsüz kesikleştirme yönteminin uygulanabilir ve verimli olduğunu göstermişlerdir (Jiang ve ark., 2009).

Tüm süreklilik arzeden özniteliklerin dönüşümü globaldir. Her türlü yerel kesikleştirme yönteminin global yönteme dönüştürülmesi sunulmaktadır. Küme analizi üzerine kurulu bir global kesikleştirme yöntemi sunulmuş ve deneysel olarak üç iyi bilinen yerel yöntemle karşılaştırılmış ve bu yöntemler globale dönüştürülmüştür. Deneyler on gerçek yaşam veri tabanından oluşmaktadır (Chmielewski ve Grzymala-Busse, 1994).

Karar ağacı ile öğrenmede kesikleştirme sürecini irdelemek önemlidir.Perner ve Trautzsch tarafından, karar ağaçlarının başlatılmasında çoklu fasıla yöntemini tarif etmektedir. İki bilinen yöntemi önerdikleri iki yöntemle sinir ağı tabanlı bir histogram (LVQ) üzerinden kaşılaştırmışlardır. Karşılaştırma doğruluk ve karar ağacının sağlamlığı açılarından yapılmıştır. IRIS alanı, uydu alanı ve OHS alanı (ovariel hyper stimulation) veri tabanları kullanılmıştır (Perner ve Trautzsch, 1998).

Kesikleştirme veri madenciliği alanında özellikle de sınıflandırma probleminde çok vermli olmaktadır. Entropi tabanlı yöntemler yereldir ve halen bilgi kaybına neden olmaktadır. Bu nedenli yeni bir entropi tabanlı algoritma sunulmuştur. Kesikleştirme başlangıç planı esnasında kesme noktaları ekleyip çıkarılabilir. Böylece kesikleştirme sürecindeki etkileşim yöntemlerine global bir özellik katılmaktadır. Deneyler, aynı kural jeneratörü C4.5 kullanmak suretiyle entropi tabanlı yöntemlere göre daha güçlü kurallar yaratabileceklerini göstermektedir (Flores ve ark., 2011).

Naive-Bayesian sınıflandırıcıya göre kesikli değişkenler ile süreklilik arz eden değişkenlerin Dirichlet öncelikleri bulunmaktadır. Bunların en önemlisi, sade Bayesian sınıflandırıcısının kesikleştirmede neden sonuç verdiğini açıklayan “mükemmel genelleştirme” özelliğidir. Mademki mükemmel genelleştirme Dirichlet’e

dayanmaktadır, o halde genel olarak kesikleştirmenin parametre tahminlerini bir normal dağılım gibi varsaydığını ve daha iyi yapabileceği söylenebilir. Bunlara ilave olarak entropi, ten-bin ve bin-log gibi iyi bilinen çok fazla sayıdaki kesikleştirme yöntemleri de açıklanabilir. Bu açıklamayı doğrulamak için sentezlenmiş ve gerçek yaşam veri tabanları kullanarak deneyler tasarlanmıştır ve iyi bilinen yöntemlerin yanı sıra analizleri süreklilik arz eden değişkenleri test verileri üzerinden kesikleştirerek tembel kesikleştirme yöntemlerine doğru götürmüştür. Dirichlet varsayımı tembel yöntemlerin başarılı kesikleştirme yöntemleri kadar performans gösterebileceğini öngörmektedir (Fayyad ve Irani, 1993).

Böylece kesikleştirme sürecindeki etkileşim yönteme global bir özellik katmaktadır. Kesikleştirme veri madenciliği alanında özellikle de sınıflandırma probleminde çok verimli olmaktadır. Entropi tabanlı yöntemler yereldir ve halen bilgi kaybına neden olmaktadır. Bu nedenle yeni bir entropi tabanlı algoritma sunulmuştur. Kesikleştirme başlangıç planı esasında kesme noktaları eklenı çıkarılabilir. Böylece kesikleştirme sürecindeki etkileşim yöntemine global bir özellik katılmaktadır. Deneyler, aynı kural jeneratörü C4.5 kullanmak suretiyle entropi tabanlı yöntemlere göre daha güçlü kurallar oluşabileceğini göstermektedir (Hsu ve ark., 2003).

Kesikleştirmenin Naive-Bayes öğrenmedeki gerçek olasılık yoğunluğunu kullandığı koşullar oluşturulmuştur. Farklı kesikleştirme tekniklerini kullanmak sınıflandırma yönüne etki edebilir ve varyans değişebilir. Naive-Bayes sınıflandırmasındaki hatayı kesikleştirme yönünü yöneterek, kesme sayısı ile her bir aradaki çalışma durumları düzenlererelr etkin bir biçimde azaltılabilir. Dolayısıyla kesikleştirme yönü ve varyansını verimli bir şekilde yönetebilen iki güdümsüz ve verimli kesikleştirme yöntemini bir başka deyişle oransal kesikleştirme ve sabit frekans kesikleştirmesini önerilmiştir. Bu yeni teknikleri dört Naive-Bayes sınıflandırıcısı ile değerlendirip istatistiksel olarak önemli frekans ve naive-Bayes sınıflandırıcılarının veri üzerinde çalıştırılmasıyla mevcut kesikleştirme yöntemleri üzerinden çalışan sınıflandırmalara göre daha düşük sınıflandırma hataları elde edilebilir (Yang ve Webb, 2009).

Değişken Hassasiyeti Kaba Kümeleri (VPRS) modeli, bilgi elde etmek üzere uygulandığında veri madenciliği uçun güçlü bir araç olsada maalesef süreklilik arzeden özniteliklere sahip gerçek yaşam sınıflandırma işleri için kullanılamaz. Değiştirilmiş Chi2 algoritması Chi2 algoritması üzerinde yapılan düzenlemelerden birisidir ve bu algoritmadaki uyumsuzluk kontrolü Kaba Setler Teorisi (RST)'den alınan benzerlik

algoritması ile değiştirilmiştir. Bu algoritma, özgürlük dereceleri etkisini dikkate almaktadır. Bununla birlikte, kontrol altındaki bir belirsizlik derecesiyle veya yanlış sınıflandırma hatasıyla yapılmış bir sınıflandırma, RST'nin alanı dışındadır. Bu algoritma aynı zamanda varyans etkisini iki birleştirilmiş arada dikkate almamaktadır. Bu çalışmada ise , bu iki sorunu aşmak üzere genişletilmiş Chi2 algoritması adıyla yeni bir algoritma önerrilmiştir. See5 yazılımını çalıştırma suretiyle önerdikleri algoritma orijinal ve düzenlenmiş Chi2 algoritmalarına göre daha iyi bir performans sergilemektedir (Su ve Hsu, 2005).

Bulanık kesikleştirme ise, her biri bir öznitelik değerine karşılık gelmek üzere sadece grup numarası ile belirtilmiş belirsiz keskinleştirmenin aksine üyelik numarası, grup numarası ve ilgi ile belirtilmektedir. Bu yaklaşımın sınıflandırma hassasiyeti cinsinden belirsiz keskinleştirmeye göre getirisi, deneysel olarak birbirini kesen veri kümelerinin kaba bir veri seti sınıflandırıcısına veri olarak girilmesi halinde deneysel olarak gösterilmiştir. Önerilen yöntemin verimliliği, ham (kesikleştirilmemiş) bilginin veri olarak girildiği çok katmanlı algılayıcıda da gözlenebilir(Roy ve Pal, 2003).

Kaba Veri Setleri Teorisi, tıp, mühendislik ve finans alanlarında uygulandığında güçlü bir veri madenciliği aracıdır. Bununla birlikte süreklilik özelliği bulunan sınıflandırma işlemlerinde gerçek dünyaya uygulanamaz. ChiBirleştir, bu alanda çokça kullanılan bir kesikleştirme yöntemi olmuştur. Chi2 algoritması, ChiBirleştir algoritmasının bir düzenlemesidir. Kesikleştirme sürecini uyumsuzluk oranı katarak otomatikleştirir ve önem seviyesini otomatik olarak belirler. Buna ilave olarak ChiBirleştir algoritması uygulama alanlarını genişletmek üzere özellik seçiminde daha hassas bir özellik barındırır. Bu konuda ne ChiBirleştir ne de Chi2 algoritmaları birleştirme kriterleri arasında uyumsuzluğu dikkate alır. Ayrıca Chi2 algoritmasının kullanıcı tanımlı uyumsuzluk oranı, üst üste birleştirmeye neden olacak şekilde kesikleştirme sürecine de uyumsuzluk getirir. Bu iki sorunun üstesinden gelmek üzere değiştirilmiş Chi2 algoritması adı altında yeni bir kesikleştirme yöntemi önerilmiştir. Yapılan karşılaştırmalı çalışmalar, değiştirilmiş Chi2 algoritmasının orijinal Chi2 algoritması ile Kaba Veri Setleri Teorisine göre çalışan herhangi bir kesikleştirme yönetimine göre daha iyi performansa sahip olduğunu göstermektedir (Shen ve Tay, 2001).

Parkinson hastalığının teşhisinde kesikleştirmenin etki analizi yaparken antropi tabanlı kesikleştirme yöntemi ile C4.5 karar ağacı ve naive-Bayes sınıflandırma yöntemleri kullanılmıştır. Parkinson hastalığı tanılamasında daha önceki yöntemlerden

hiçbirisi kullanılmamıştır. Parkinson hastalığı veri setleri tasnif edilmiş ve sonuçlar karşılaştırıldığında, keskinleştirme yöntemi kullanılması halinde Parkinson hastalığının teşhisindeki başarının %4.1'den %12.8'e çıktığı gözlemlenmiştir (Kaya ve ark., 2011).

Majak ve arkadaşlarının çalişmasında mevcut çalışmada FG ışınlarında Haar dalgacıkları için kesikleştirme yönteminin (HWDM) adaptasyonuna ve bu işlemin doğruluğuna yer verilmiştir. Çok geniş alandaki komopozit ve nanoyapı sınıflarını kapsayan diferansiyel denklemler için yakınsama analizi yapılmıştır. Buna ilişkin hata limitleri belirlenmiştir. Richardson dış değer yöntemi uygulanmak suretiyle HWDM yakınsama düzeninin ikiden dörde çıktığı gösterilmiştir. FGM ışınını model problem olarak dikkate alınmasıyla sayısal örneklerle teorik tahminler doğrulanmıştır. HWDM uygulamak suretiyle elde edilen sonuçlar sonlu farklar yöntemi (FDM) sonuçlarıyla karşılaştırılmıştır (Majak ve ark., 2015).

3. MATERYAL VE YÖNTEM

3.1. Veri Setleri

İris: 1988 yılında R.A. Fisher ve Michael tarafından hazırlanmıştır. Bu veri seti iris çiçeklerinin türlerini belirtmektedir. Bu veri setinde çanak yaprağı uzunluğu (sepal length), çanak yaprağı genişliği (sepal width), taç yaprağı uzunluğu (petal length) ve taç yaprağı genişliği (petal width) olmak üzere 4 adet nitelik bulunmaktadır. Bu niteliklerin değerlerine göre iris çiçekleri, iris sedosa, iris versicolour, iris virginica olmak üzere 3 adet sınıfa ayrılmıştır. 150 adet veri bulunmaktadır (Fisher, 1936).

Denge (Balance): Siegler, R. S. (1976) tarafından rapor edilen psikolojik deneyleri modellemek için üretilmiştir. 625 adet veri bulunmaktadır. Nitelik sayısı 4'tür (Hume, 1994-04-22).

Muz (Banana): Örneklerin, muz şeklindeki birkaç kümeye ait olduğu yapay bir veri kümesidir. Sırasıyla x ve y eksenine karşılık gelen At1 ve At2 olmak üzere iki nitelik vardır. Sınıf etiketi (-1 ve 1), veri kümesindeki iki muz şeklinden birini temsil eder.

Şarap (Wine): 1991 yılında M. Forina ve ark., Parvus tarafından hazırlanmıştır. Bu veri setinde alkol, malik asit, kül, kül alkalinitesi, magnezyum, toplam fenoller, flavonoidler, inflavonoid fenoller öznitelikleri bulunmaktadır. 178 adet veri bulunmaktadır. Nitelik sayısı 13'tür (Aeberhard, 1991) .

Ecoli: 1996 yılında Osaka Üniversitesi Kenta Nagai Moleküler ve Hücresel Biyoloji Enstitüsü tarafından hazırlanmıştır. 336 adet veri bulunmaktadır. Nitelik sayısı 8'dir (Horton, 1996).

Cam (Glass): 1987 yılında B. German Merkezi Araştırma Kuruluşu tarafından hazırlanmıştır. 204 adet veri bulunmaktadır (Vina Spiehler, 1987).

İyonosfer (Ionosphere): 1989 yılında Johns Hopkins Üniversitesi Uygulamalı Fizik Laboratuvarı Uzay Fiziği Grubu tarafından hazırlanmıştır. 350 adet veri bulunmaktadır. Nitelik sayısı 34'dür (Sigillito, 1989).

Sihir (Magic): Çek Cumhuriyeti CR Bilgisayar Bilimi Enstitüsü Major Atmospheric Gamma Imaging Cherenkov Telescope projesi (MAGIC) dâhilinde 2007 yılında R. K. Bock tarafından hazırlanmıştır. 19020 adet veri bulunmaktadır. Nitelik sayısı 11'dir (Savicky, 2007) .

Yeni Tiroid (Newthyroid): 1987 yılında Ross Quinlan tarafından hazırlanmıştır. 7200 adet veri bulunmaktadır. Nitelik sayısı 21'dir (Quinlan, 1987).

Pima: Veri setleri farklı özellikli tıbbi değişkenlerden ve hedef değişkenin sonucundan oluşur, belirleyicilere hamilelik sayıları, BMI, insülin seviyesi, yaş ve benzerleri dâhildir.

Ring: Los Angeles, Kaliforniya Üniversitesi David Draper tarafından 1993 yılında hazırlanmıştır 23 adet veri bulunmaktadır. Nitelik sayısı 4'dür (Draper 1993) .

Image Segmentation: 1990 yılında Massachusetts Üniversitesi Vision Group tarafından hazırlanmıştır. Örnekler, 7 dış mekân görüntüsünden oluşan bir veri tabanından rastgele çizilmiştir. Görüntüler, her piksel için bir sınıflandırma oluşturmak üzere elden geçirilmiştir. Her örnek bir 3x3 piksel bölgesidir. 2610 adet veri bulunmaktadır. Nitelik sayısı 19'dur (Group, 1990).

Yeast: 1996 yılında Osaka Üniversitesi Kenta Nakai Moleküler ve Hücresel Biyoloji Enstitüsü tarafından hazırlanmıştır. 1484 adet veri bulunmaktadır. Nitelik sayısı 8'dir (Creator ve Maintainer, 1996).

3.2. Yöntemler

3.2.1. Sınıflandırma işlemi

Sınıflandırma, belirli bir girdiye dayanan belirli bir sonucu tahmin etmekten oluşur. Sonucun tahmin edilmesi için, algoritma, bir takım nitelikler ve genellikle hedef veya tahmin niteliği olarak adlandırılan ilgili sonucu içeren bir eğitim setini işler. Algoritma, sonuçları tahmin etmeyi mümkün kılacak nitelikler arasındaki ilişkileri keşfetmeye çalışır. Daha sonra, algoritmaya tahmin seti adı verilen, önceden bilinmeyen bir veri seti verilir (Han ve ark., 2011). Algoritma girdiyi analiz eder ve bir tahmin üretir. Tahmin doğruluğu, algoritmanın ve veri setinin "iyi" olup olmadığını tanımlar. Örneğin, bir medikal veri tabanında, eğitim setinde önceden kaydedilmiş ilgili hasta bilgileri olur; tahmin özelliği, hastanın bir kalp problemi olup olmadığıdır. Aşağıdaki Çizelge 3.1 ve Çizelge 3.2, bu tür bir veri tabanının eğitim ve tahmin setlerini göstermektedir (Han ve ark., 2011).

Çizelge3. 1 Medikal veri tabanı eğitim seti

Yaş	Kalp atış hızı	Kan basıncı	Kalp problemi
65	78	150/70	Evet
37	93	112/76	Hayır
71	67	108/65	Hayır

Çizelge3. 2 Medikal veri tabanı tahmin seti

Yaş	Kalp atış hızı	Kan basıncı	Kalp problemi
43	98	147/89	?
65	83	106/63	?
84	77	150/65	?

Literatürde mevcut bilgi gösterimi türleri arasında sınıflandırma, normal olarak bilgiyi ifade etmek için tahmin kurallarını kullanır. Tahmin kuralları, IF-THEN kuralları biçiminde ifade edilir; burada IF kısmı, koşulların bir birleşiminden oluşur ve sonuçtaki kural (SON bölüm), tatmin eden bir öğenin belirli tahminleri öznitelik değerini öngörür. Yukarıdaki örneği kullanarak, eğitim kümesindeki ilk satırı öngören bir kural Denklem 3.1 gibi temsil edilebilir:

IF (Age=65 AND Heart rate>70) OR (Age>60 AND Blood pressure>140/70) THEN Heart problem=yes (3.1)

Çoğu durumda, tahmin kuralı, yukarıdaki örneğe kıyasla çok daha büyüktür. Birleşim sınıflandırma için güzel bir özelliğe sahiptir; OR'lar ile ayrılmış her durum, nitelikler arasındaki ilişkileri yakalayan daha küçük kurallar tanımlar. Bu küçük kurallardan herhangi birini yerine getirmek sonuçta tahmin anlamına gelir. Her küçük kural, nitelikler arasındaki ilişkileri daraltmayı kolaylaştıran AND'ler ile oluşturulmuştur (Khoo ve ark., 1999).

Tahminlerin ne kadar iyi yapıldığı, tahminlerin toplam sayısına karşı gelen doğru tahminlerin yüzdesiyle ölçülür. İyi bir kuralda, isabet oranının tahmini öznitelikten daha büyük olması gerekir. Başka bir deyişle, algoritma Seattle'da yağmur yağmayı öngörmeye çalışıyorsa ve zamanın %80'inde yağmur yağarsa, algoritma her an yağış tahmin ederek kolayca %80'lik bir isabet oranına sahip olabilir. Bu nedenle, %80 herhangi bir algoritmanın bu durumda elde etmesi gereken taban tahmin hızıdır. En uygun çözüm, %100 tahmini isabet oranına sahip olan bir kuraldır. Bu nedenle, bazı çok özel problemler haricinde, tanımlama yoluyla sınıflandırma ancak yaklaşım algoritmaları ile çözülebilir.

3.2.1.1. Bilgi kazancına göre sınıflandırma (Karar ağacı ID3)

Makine öğreniminde, çoğu çalışma bilgi teorisine dayanmaktadır. Yaptığı işi Hunt'la ilişkilendiren ID3 Quinlan yöntemini Karar Ağacının Quinlan 1979 alıntılamak

genel bir yaklaşımdır (Du ve Zhan, 2002). Quinlan, 80'li yılların ikinci yarısında, sistemin davranışını iyileştirmek için bir buluşsal yöntem öneren çok sayıda yayınla çok aktif bir yöntem olmuştur (Han ve ark., 2011).

Sınıflandırma yöntemleri, nesneleri betimleyen belirli özelliklere ait sınıfları belirlemeyi amaçlamaktadır. İnsan faaliyetleri sınıflandırma yönremlerine geniş bir yelpazede ve özellikle otomatik karar vermede yarar sağlarlar. Karar ağaçları, gözetimli öğrenmenin çok etkili bir yöntemidir. Bir veri kümesinin tahmin edilecek değişken açısından mümkün olduğunca aynı yapıda olan gruplara ayrılmasını amaçlar. Giriş olarak bir seri veri kümesini alır ve her bitiş düğümünün (yaprak) bir karar (bir sınıf) ve son olmayan her düğümün (iç) bir testi temsil ettiği bir oryantasyon diyagramına benzeyen bir ağaç çıkarır. Her yaprak, kökten yaprağa tüm test yollarını doğrulayan bir veri sınıfına ait olma kararıdır.

Ağaç basit ve teknik olarak kullanımı kolaydır. Aslında, test edilecek değişkenlerin olasılıklarına uyarlanmış bir ağaç elde etmek daha iyidir. Çoğunlukla dengelenmiş ağaçlar iyi bir sonuç vermektedir. Eğer bir alt ağaç sadece benzersiz bir çözüme yol açabilirse, tüm alt ağaç basit sonuçlara indirgenebilir. Bu süreci basitleştirir ve nihai sonucu değiştirmez. Ross Quinlan bu tür karar ağaçlarında çalışmıştır. Shannon teorileri ID3 ve C4.5 gibi sınıflandırma algoritmalarının temelidir. Entropi Shannon en çok bilinen ve en çok uygulananlardandır. İlk önce, bir olay tarafından sağlanan bilgi miktarını tanımlar: bir olayın olma olasılığı ne kadar yüksekse o kadar çok bilgi verir.

3.2.1.2. Shannon Entropi

Genel olarak, bir olasılık dağılımı P = (p_1, p_2, ..., p_n) ve bir S örneği verilirse, P'nin entropisi olarak da adlandırılan bu dağılım tarafından taşınan bilgi Denklem3.2'deki gibi verilir:

$$Entropi(p) = -\sum_{i=1}^{n} p_i \times log(p_i) \quad (3.2)$$

3.2.1.3. Bilgi kazanımı

Tüm örnekler için sınıfların karıştırılma derecesini ağacın herhangi bir konumunu ölçülmesine izin veren fonksiyonlar bulunmaktadır. Ağaç yapısını oluşturmak için her özelliğin bilgi kazanımını hesaplamak gerekmektedir. Bilgi kazanımı denkleme Denklem3.3'e göre hesaplanmaktadır (Du ve Zhan, 2002).

$$Gain(P,T) = Entropi(p) - \sum_{j=1}^{n} \left(p_j \times Entropie(p_j)\right) \quad (3.3)$$

Burada değerler (Pj), T özniteliği için olası tüm değerlerin kümesidir. Öznitelikleri önemine göre sıralamak ve bu sıraya göre karar ağacını oluşturmak gerekir. En yüksek bilgi kazanımına sahip olan öznitelik ağacın kökünü temsil eder.

3.2.1.4. İstatistiksel hesaplamalara göre sınıflandırma

Bayes sınıflandırıcıları istatistiksel sınıflandırıcılardır. Belli bir örneğin belirli bir sınıfa ait olma ihtimali gibi sınıf üyelik olasılıklarını öngörebilirler. Bayes sınıflandırıcısı Bayes teoremine dayanmaktadır. NB sınıflandırıcıları, belirli bir sınıfa bir nitelik değerinin etkisinin diğer özelliklerin değerlerinden bağımsız olduğunu varsaymaktadır. Bu varsayıma sınıfsal şartsız bağımsızlık denir. Hesaplamayı basitleştirmek için yapılır ve bu anlamda " Naive" olarak kabul edilir (Witten ve ark., 2016).

3.2.1.5. Bayes teoremi

$X = \{x_1, x_2, \ldots, x_n\}$ örnekleri, n niteliğinde yapılan bir dizi değerleri temsil eden bir örnek olsun. Bayesian terimleriyle X, "evidence" olarak kabul edilir. H hipotezi olsun ve X verisi belirli bir sınıf C'ye ait olsun. Sınıflandırma problemleri için amaç P(H | X), " evidence " göz önüne alındığında H hipotezinin tutma olasılığını belirlemektir. Bir başka deyişle, X'in öznitelik tanımı bilindiği için, X örneğinin C sınıfına ait olma ihtimalini hesaplıyoruz. P(H | X), H'in X üzerinde koşullandırılmış bir posteriori olasılığıdır. Örneğin, veri örneklerimizin yaş ve gelir olduğunu ve bu örneğin geliri 40.000 TL olan 35 yaşındaki bir müşteriye ait olduğunu varsayalım. H, müşterinin bir bilgisayar satın alacağı hipotezi olduğunu varsayalım. Sonra P (H | X), müşterinin yaşı ve geliri bilindiği için X müşterisinin bir bilgisayar satın alması ihtimalidir (Han ve ark., 2011).

Buna karşın, P(H), H'nin olasılığıdır. Örnek için, bu herhangi bir müşterinin yaş, gelir veya başka bilgilerden bağımsız olarak bir bilgisayar satın alma olasılığıdır. Bir posteriori olasılığı P(H | X), X'den bağımsız bir olasılık P (H) 'ya göre daha fazla bilgiye (müşteri hakkında) dayalıdır.

Benzer şekilde, P(X|H) X'in H üzerinde koşullandırılmış bir posteriori olasılığıdır. Yani, müşterinin bir bilgisayar satın alacağı bilirdiği için, bir müşteri olan X'in 35 yaşında ve 40.000 TL kazanma olasılığıdır.

P(X), X'in bir priori olasılığıdır. Örnek, müşterilerden birinin 35 yaşında ve 40.000 TL kazanma ihtimali verilmektedir. Bayes teoremine göre, P (H | X) olasılığı P (H), P (X | H) ve P (X) olarak ifade edilebilir (Denklem3.4).

$$P(H|X) = P(X|H)\, P(H) / P(X) \quad (3.4)$$

3.2.1.6. Naive Bayes Sınıflandırıcı

NB sınıflandırıcısının çalışma şekli aşağıdaki gibidir (Murphy, 2006):

Adım 1: T, her biri sınıf etiketlerine sahip bir örnek eğitim seti olsun. K sınıfları, $C_1, C_2, \ldots C_k$. Her örnek, n boyutlu bir vektörle temsil edilir $X = \{x_1, x_2, \ldots, x_n\}$, ve son olarak $A_1, A_2, \ldots, A_n$, n niteliklerin değerlerini temsil etmektedir.

Adım 2: X bir örnek verildiğinde, sınıflandırıcı, X'in şarta bağlı olarak X'in en üst düzeyde bir posteriori olasılığa sahip olan sınıfına ait olduğunu tahmin edecektir. Yalnızca Denklem 3.5 göre X'in C_i sınıfına ait olduğu tahmin edilir P(Ci|X) > P(Cj |X) Böylece, P(Ci|X)'i maksimize eden sınıf bulunur. P(Ci | X) 'in en büyük hale getirildiği Ci sınıfına maksimum posteriori hipotezi denir.

$$P(Ci|X) = P(X|Ci) / P(X) * P(Ci) \quad (3.5)$$

Adım 3: Tüm sınıfların P (X) aynı olduğu için yalnızca P (X | Ci) P (Ci) en büyük hale getirilmelidir. Sınıf olasılıkları, P (Ci) bilinmiyorsa, sınıfların eşit derecede muhtemel olduğu, yani P (C1) = P (C2) = …. = P (Ck) olduğu varsayılır ve bu nedenle P (X | Ci) maksimize edilir.

Adım 4: Birçok özelliğe sahip veri kümeleri göz önüne alındığında, P (X | Ci) hesaplamak hesaplama açısından pahalı olurdu. P (X | Ci) P (Ci) değerinin hesaplanmasında hesaplanmayı azaltmak için sınıfsal şartlı bağımsızlığın naif varsayımı yapılır. Bu, örneğin sınıf etiketiyle birlikte, özelliklerin değerlerinin koşullu olarak birbirinden bağımsız olduğunu varsayar. Matematiksel olarak Denklem3.6'daki gibidir,

$$p(X/C_i) \approx \prod_{k=1}^{n} P(X_k/C_i) \cdot \quad (3.6)$$

Adım 5: X örneğinin sınıf etiketini tahmin etmek için P (X | Ci) P (Ci) her Ci sınıf için değerlendirilir. Sınıflandırıcı, X'in sınıf etiketinin, yalnızca ve P (X | Ci) P (Ci) 'yi maksimize eden sınıf olması durumunda, Cı olduğu öngörüsünde bulunur.

3.2.2. Ayrıklaştırma işlemi

Ayrıklaştırma işlemi, nicel veriyi nitel veriye yani sayısal özniteliklere, sonlu sayıda aralıklarla ayrı veya nominal özniteliklere dönüştürerek, sürekli bir alanın birbirine geçmeyen bir bölümünü elde eder (Dougherty ve ark., 1995a). Ardından her

aralık ile sayısal ayrık bir değer arasında bir ilişki kurulmuştur. Uygulamada ayrıklaştırma, büyük bir spektrumdaki sayısal değerlerden oluşan veriyi, harici değerlerin büyük ölçüde küçülmüş bir alt kümesine eşleştirdiğinden, veri azaltma yöntemi olarak görülebilir. Sayısal veriler üzerinde ayrıklaştırma işlemi gerçekleştirildikten sonra VM bu verileri ayrık veriler olarak işlemlere tabi tutar. Birçok mevcut VM algoritması, ayrık veriler kullanarak öğrenmelerini gerçekleştirebilir, fakat gerçek dünya uygulamaları genelde sürekli veriler içerir (Garcı´a ve ark., 2013). Bu sayısal verilerin, bu gibi VM algoritmalarında kullanılmadan önce ayrık hale getirilmesi gerekmektedir. Literatürde araştırmacılar tarafından geliştirilen birden fazla ayrıklaştırma yöntemleri bulunmaktadır (Kotsiantis ve Kanellopoulos, 2006b). Genelde bu yöntemler denetimli ve denetimsiz diye iki ana gruba ayrılmaktadır. Dolayısıyla bu çalışmada denetimsiz ayrıklaştırma yöntemlerinden olan EG ile EF yöntemleri bir örnek üzerinde detaylı bir şekilde incelenmektedir. Diğer taraftan entropi temelli denetimli olan ID3 yöntemi detaylı olarak örnek üzerinde açıklanmıştır.

3.2.2.1. Ayrıklaştırma Yöntemlerinin Özellikleri

Ayrıklaştırma yöntemlerinin kategorize edilmesini sağlamak için çeşitli eksenler tanımlanmıştır. Bu bölümde bunlar açıklanmakta ve incelenerek ana hususları ve ilişkileri vurgulanmaktadır. Önerilen taksonomi şu özelliklere dayanmaktadır:

• Statik ve Dinamik: Bu özellik, ayrıklaştırıcının öğrenmeye ilişkili olarak çalıştığı andaki bağımsızlığa atıfta bulunmaktadır. Dinamik bir ayrıklaştırıcı, öğrenme modeli oluştururken hareket eder, böylece öğrenme modelinde kullanılan bazı işlemler ilişkili olarak kullanılır. Aksi takdirde, statik bir ayrıklaştırıcı öğrenme görevinden ve öğrenme algoritmasından bağımsızdır. Sayısal veriyle uğraşırken, dinamik ayrıklaştırıcıların çoğunun gerçekten alt bölümleri veya VM algoritmaları aşamaları olması nedeniyle hemen hemen bilinen tüm ayrıklaştırıcılar statiktir. Bilinen dinamik tekniklerden bazı örnekler ID3 ayrıklaştırcı yöntemidir.

• Tek Değişkenli ve Çok Değişkenli: Çok parçalı teknikler, aynı zamanda 2B ayrıklaştırma olarak da bilinirler, aynı anda tüm kesme noktalarını tanımlamak veya en iyi kesme noktasını belirlemek için bütün özellikleri dikkate alırlar. Ayrıca, yüksek nitelikli ilişkileri kullanarak diğer niteliklerle olan etkileşimleri inceleyen bir öznitelik ayırdedebilirler. Tek değişkenli ayrıklaştırıcılar ise bir kereye mahsus olmak üzere, tek seferde tek bir öznitelikle çalışmaktadır; bunlar, nitelikler arasında bir sipariş oluşturulduktan sonra ortaya çıkar ve sonuçta ortaya çıkan ayrıştırma şeması, her

öznitelikte sonraki aşamalarda değişmeden kalır. Yakın geçmişte, çok değişkenli ayrıklaştırıcıların geliştirilmesinde ve çoklu öznitelikler arasındaki yüksek etkileşimlerin bulunduğu tümdengelimi öğrenmede, tek değişkenli ayrıklaştırıcıların karmaşık sınıflandırma sorunlarını önlemede etkili oldukları için ilgi artmıştır.

• Denetimli ve Denetimsiz: Denetimsiz ayrıklaştırıcılar sınıf etiketi göz önüne almazken denetlenen ayrıklaştıcılar sınıf etiketini göz önüne alır. Denetlenen tipin sınıf niteliğini dikkate alma şekli, girdi özellikleri ve sınıf etiketleri arasındaki etkileşime ve en iyi kesme noktalarını (entropi, karşılıklı bağımlılık, vb.) belirlemek için kullanılan sezgisel yöntemlere bağlıdır. Literatürde önerilen çoğu ayrıklaştırma denetimlidir ve teorik olarak, sınıf bilgisi kullanılarak, her özellik için otomatik olarak en iyi aralık sayısını belirler. Ayrıklaştırma denetlenmezse, denetlenen görevler üzerine uygulanamayacağı anlamına gelmez. Fakat denetlenen bir ayrıştırıcı yalnızca denetlenen VM sorunları üzerine uygulanabilir. Denetimsiz ayrıklaştırmalar EG ve EF gibi ayrıklaştırma yöntemleridir. Entropi tabanlı olan ID3 ayrıklaştırma yöntemi ise denetimli ayrıklaştırma yöntemidir.

• Bölme ile Birleştirme: Bu, yeni aralıklar oluşturmak veya tanımlamak için kullanılan yöntemi ifade eder. Bölme yöntemleri, tüm olası sınır noktaları arasında kesme noktası oluşturur ve alanı iki aralıkla bölüştürür. Buna karşılık, birleştirme yöntemleri önceden tanımlanmış bir bölümle başlar ve her iki bitişik aralığı karıştırmak için bir aday kesme noktasını kaldırır. Bu özellikler sırasıyla üst-alt ve alt-üst ile yüksek derecede ilişkilidir (bir sonraki bölümde açıklanmıştır). Hiyerarşik bir ayrıklaştırma yapısına göre, yukarıdan aşağıya veya aşağıdan yukarıya ayrıklaştırıcıların sürecin aşamalı olduğunu (daha sonra açıklanacağı) varsaymaları dışında onların arkasındaki fikir çok benzerdir. Aslında, bölme veya bir kerede birden fazla aralık birleştirme işlemine dayanan ayrıklaştırıcılar olabilir. Ayrıca, bazı ayrıklaştırıcılar, bölünmeler yerine çalışma zamanında birleşmeler yapabilmeleri nedeniyle birleştirme olarak kabul edilebilir.

• Global ile yerel: Karar vermek için, bir ayrıklaştırıcı ya özellikte mevcut tüm verileri isteyebilir ya da yalnızca kısmi bilgileri kullanabilir. Ayrıklaştırıcı, yalnızca bölme kararını yerel bilgilere (veri kümesinin özellikleri ve sınıf bilgisi) dayandırdığında yerel olarak söylenir. Yaygın olarak kullanılan yerel teknikler ID3 yöntemidir. Bazıları üstten aşağıya bölme ve tüm dinamik teknikler temelli bazıları hariç, yerel olanlar azdır. EF ve EG gibi ayrıklaştırma yöntemleri kesme noktalarının kararını kullanıcılardan aldığı için global olarak bilinmektedir.

• Doğrudan artımlı: Doğrudan ayrıklaştırıcılar aralığı k aralığına aynı anda böler ve k değerini belirlemek için ek bir ölçüt gerekir. Bunlar sadece bir adımlı ayrıklaştırma yöntemleri içermez, aynı zamanda işlemlerinde birkaç aşamayı gerçekleştiren ve her adımda birden fazla kesme noktası seçen ayrıklaştırıcılar içermektedir. Buna karşılık, artımlı yöntemler basit bir ayrıklaştırma ile başlar ve iyileştirme sürecinden geçer ve bunu durdurma zamanını bilmek için ek bir ölçüt gerekir. Her adımda, kesim noktası olarak kullanılacak en iyi aday sınırını bulurlar ve daha sonra kalan kararlar buna göre yapılır. Artımlı ayrıklaştırıcılar hiyerarşik ayrıklaştırıcılar olarak da bilinir. Her iki ayrıklaştırıcı türü de literatürde yaygındır, ancak artan ve denetlenenler arasında genellikle daha belirgin bir ilişki vardır.

• Veri ön işleme: Bu kategori bir değerlendirme önleminin yokluğuna atıfta bulunmaktadır. Belirli bir sayıda kutu oluşturarak bir niteliği ayırmak için en basit yöntemdir. Her bir depo önsel olarak tanımlanır ve özellik başına belirli sayıda değer tahsis eder. Sık kullanılan binning yöntemleri EW, EF'dir.

3.2.2.2. Eşit Genişlik Ayrıklaştırma Yöntemi

EG ayrıklaştırma yöntemi denetlenmeyen ayrıklaştırma yöntemidir, ayrıklaştırma işlemini gerçekleştirirken sınıf bilgisini kullanmaz. Bu ayrıklaştırma yönteminde sürekli olan değerler belirli aralıklar ile ayrık değerlere dönüştürülür. Fakat bu yöntemde k dediğimiz aralık sayıları kullanıcı tarafından verilmektedir. K aralık değerine göre aralıklar elde edilir. Denklem3.7 2ve 3.8'de sürekli bir dizinin elemanları için interval ve aralıkların fonksiyonları verilmektedir (Hacibeyoglu ve ark., 2011).

$$interval = \frac{a_{max} - a_{min}}{k} \tag{3.7}$$

$$Aralık = a_{min} + (i * interval) \tag{3.8}$$

Burada a_{max} dizideki en büyük değeri, a_{min} dizideki en küçük değeri ve k ise parça sayısını temsil etmektedir. Aralık denkleminde ise dizideki en küçük değere i aralık indeksi ve aralık değeri ile toplanarak dizinin aralıkları bulunur. EG Aralık Çizelge 3.3'te gösterilmiştir T sıcaklık, C sınıf değeridir. A ise, T'nin ayrıklaştırılmış değeri olan T'nin sıralanmış değeridir. Bu algoritmanın girdi verisi, A = {a_0, a_1, a_2,… ..

a_{n-1}, a_n} özniteliğinin sürekli değerlerine sahip dizidir ve k, parçaların sayısıdır (k>0). Çıktısı ise ayrık değerlere sahip verilerdir.

Çizelge 3.3 Eşit Genişlik Ayrıklaştırma yöntemi için örnek

I	1	2	3	4	5	6	7	8	9	10
T	75	80	76	86	70	60	50	85	65	81
C	no	no	yes	yes	yes	no	yes	no	yes	yes
A	50	60	65	70	75	76	80	81	85	86
a_{min}=50, a_{max}=86 k=3 Aralık=(86-50)/3=12 Limit=50+12=62 0 = [50, 62); 1 = [62, 74); 2 = [74, 86]										
D	2	2	2	2	1	0	0	2	1	2

Çizelge 3.3'te I dizideki eleman sırasını, T dizi adını, C ise T dizisindeki elemanların ait olduğu sınıfı, A ise T dizisindeki elemanların sıralı hali ve D ise dizideki değerlerin ayrık değerlerini temsil eder.

3.2.2.3. Eşit Frekans Ayrıklaştırma Yöntemi

EF ayrıklaştırma yöntemi denetlenmeyen ayrıklaştırma yöntemlerinin bir örneğidir. Bu ayrıklaştırma yönteminde sürekli özelliğine sahip olan dizi elemanları her aralıktaki eleman sayısına göre yapılmaktadır. Her aralıktaki eleman sayısı Denklem3.9 ile hesaplanmaktadır (Hacıbeyoğlu ve Ibrahim, 2016).

$$Her\ aralıkta\ eleman\ sayısı = \frac{n}{k} \tag{3.9}$$

Burada n dizideki eleman sayısını ve k ise aralık sayısını temsil etmektedir. Ayrıklaştırma yöntemini gerçekleştirmek için dizideki en büyük ve en küçük değerin bulunması gerekmektedir. EG'lı ayrıklaştırma algoritmasında ayrıklaştırılmış öznitelik için minimum ve maksimum değerler belirlenir ve daha sonra tüm değerler artan düzende sıralanır ve sürekli değerler her aralıkta yaklaşık n / k veri örnekleri içerecek şekilde k aralıklarına ayrılır. A = {a_0, a_1, a_2, ... a_{n-1}, a_n} sürekli değerlere sahip olan dizili bir veri kümesini ve n bu dizideki eleman sayısını ifade etmektedir k sayıları [1], [3] , [5] ve [7]. Orantılı k- aralıklı ayrıklaştırma yönteminde, aynı değere sahip veri örnekleri aynı aralığa yerleştirilmelidir, bu nedenle tam olarak k EF aralıkları üretmek

her zaman mümkün değildir [7]. Çizelge 3.4'de örnek olarak verilmiştir. Örnekte T sıcaklık, C sınıf değeridir. A T'nin sıralı bir değeri, D ise T'nin ayrıklaştırılmış değerleridir. Bu algoritmanın girdi verisi, A = {a0, a1, a2,… .. an-1, an} özniteliğinin sürekli değerlerine sahip dizidir ve k, parçaların sayısıdır (k>0). Çıktısı ise ayrık değerlere sahip verilerdir.

Çizelge3 .4 Eşit Frekans Ayrıklaştırma yöntemi örneği

I	1	2	3	4	5	6	7	8	9	10
T	75	80	76	86	70	60	50	85	65	81
C	no	no	yes	yes	yes	no	yes	no	yes	yes
A	50	60	65	70	75	76	80	81	85	86
a_{min}=50, a_{max}=86 n=10, k=3 Aralık=10/3=3.3 0 = [50, 70); 1 = [70, 80); 2 = [80, 86]										
D	1	2	1	2	1	0	0	2	0	2

Çizelge 3.4'te I dizideki eleman sırasını, T; dizi adını, C; T dizisindeki elemanların ait olduğu sınıfı, A ise T dizisindeki elemanların sıralı halini ve D ise dizideki değerlerin ayrık değerlerini temsil eder.

3.2.2.4. Kümelenme tabanlı ayrıklaştırma

K-ortalama kümeleme yöntemi, en popüler kümeleme yöntemlerinden biridir ve küme veri noktalarına sürekli mesafe temelli benzerlik ölçümü hesapladığı için, sürekli değerli değişkenleri ayrıklaştırmak için de uygundur (Dash ve Paramguru, 2011). K-ortalama kümeleme yöntemi denetimsiz bir ayrıklaştırma yöntemidir. Bu yöntemde veri kümesinin sınıf bilgisi kullanılmadan ayrıklaştırma işlemi gerçekleştirilir. Önce kullanıcı tarafından verilen k parça sayısı kadar rastgele küme merkezleri atanır, veri kümesindeki her veri noktasının bütün kümelere olan uzaklıkları hesaplanır ve en yakın olana atanır daha sonra küme merkezleri güncellenir.

Çizelge 3 .5 K-ortalama yöntemine örnek

I	1	2	3	4	5	6	7	8
T	10	50	15	20	12	25	40	30
C	Hayır	Hayır	Evet	Evet	Evet	Hayır	Evet	Hayır
A	10	12	15	20	25	30	40	50
	K= 2 ve rastgele olarak 15, 30 küme merkezleri olarak atanmıştır							
D	1	2	1	1	1	1	2	2

Çizelge3.5'te I dizideki eleman sırasını, T dizi adını, C ise T dizideki elemanların ayıt olduğu sınıf, A ise T dizideki elemanların sıralı halı ve D ise dizideki değerlerin ayrık değerlerini temsil eder.

3.2.2.5. Entropi Temeli ayrıklaştırma yöntemi (ID3)

Entropi temelli ayrıklaştırma yöntemi denetimli bir ayrıklaştırma yöntemidir. Bu yöntem veri kümesinin sınıf bilgisini kullanarak ayrıklaştırma işlemini gerçekleştirir. Bu ayrıklaştırma yöntemi ID3 sınıflandırma algoritmasında kullanılan verilerin bilgi kazancını kullanarak aralık değerlerini belirler. Verilerin bilgi kazançlarını hesaplamak için önce verilerin belirsizliği olan entropileri hesaplanmaktadır, her özelliğin entropisi Denklem 3.10 ve 3.11'e göre hesaplanır (Hacibeyoglu ve ark., 2011).

$$E(A;S) = -\sum_{i=1}^{n} \frac{S_i}{S} E(S_i) \quad (3.10)$$

$$E(S_i) = -\sum_{i=1}^{n} P_i log_2 P_i \quad (3.11)$$

Burada, A özelliğinin S kesme noktalarının entropileri Denklem3.12'ye göre hesaplanır. P dizideki elemanların olasılığını ifade eder ve n ise dizideki aralık sayılarını temsil eder. Her aralığın entropisi hesaplandıktan sonra bilgi kazançları Denklem 3.12' ile hesaplanır.

$$bilgi\ kazancı = E(S) - E(A;S) \quad (3.12)$$

Yüksek bilgi kazancına sahip olan aralık sayısı algoritma tarafından kabul edilmektedir. Entropi tabanlı ID3 ayrıklaştırma yönteminin çalışma adımları A dizisi üzerinde açıklanmıştır: A = {a_0, a_1, a_2,…, a_n} özniteliğinin sürekli değerlerine sahip dizidir. Birinci adımda A değerini artan düzende sıralanır. İkinci adımda sınıf verileri için entropi hesaplanır. Üçüncü adımda A'ya göre en iyi ayrılma noktası belirlenir. Her bölünme noktası için Entropi'yi ve bilgi kazancını hesaplanır. Dördüncü adımda en yüksek bilgi kazancı olan ayrılma noktasını seçilir. Beşinci adımda bilgi kazanımı belirli bir eşik değerin altına düşene kadar adımlar tekrarlanır.

Çizelge 3.6 Entropi Tabanlı Ayrıklaştırma yöntemi örneği

I	1	2	3	4	5	6	7	8	9	10
T	75	80	76	86	70	60	50	85	65	81
C	no	no	yes	yes	yes	no	yes	no	yes	yes
A	50	60	65	70	75	76	80	81	85	86
Hayır = 4, Evet = 6, Olasılık Hayır Phayır = 4/10 = 0.4, Olasılık Evet Pevet= 6/10 = 0.6 Entropi (başlatma) = E (4, 6) = E (0.4, 0.6) = $-0.4*\log_2(0.4) - 0.6*\log_2(0.6)$ = 0.970. 1. Bölme <= 70 = {50, 60, 65, 70} ve> 70= {75, 76, 80, 81, 85, 86} 2. Bölme <= 76 = {50, 60, 65, 70, 75, 76} ve> 76 = {80, 81, 85, 86} Her bölme için entropi ve bilgi kazancını hesaplayın ve ardından en yüksek bilgi kazancı olan bölmeyi seçin. Entropi E = - Σ pi $\log_2$ (pi) Entropi birinci bölme için= 0.954 ve Entropi ikinci bölme için= 0.888. Bilgi Kazancı = E(S) – E(S, A) Bölme1 = 0.970 - 0.954 = 0.016 and Bölme2 = 0.970 - 0.888 = 0.082 then 0 = [50, 76); 1 = [76, 86].										
D	0	1	0	1	0	0	0	1	0	1

Çizelge 3.6'da I dizideki eleman sırasını, T; dizi adını, C; T dizisindeki elemanların ait olduğu sınıfı, A; T dizideki elemanların sıralı halini ve D ise dizideki değerlerin ayrık değerlerini temsil etmektedir.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez çalışmasında VM'de sık kullanılan EG, EF ve ID3 ayrıklaştırma yöntemleri UCI veri setleri üzerinde üç farklı sınıflandırma algoritmaları üzerinde test edilip ve test sonuçları karşılaştırılmıştır. EG, EF ve ID3 ayrıklaştırma yöntemleri visual studio C#.Net 2013 programlama dilinde kodlanmıştır, kodlar EK-1'de verilmiştir ,visual studionun testlerinin ekran görüntüsü Şekil 4.1 ve 4.2'de verilmiştir ve testleri i7 1.5GHz işlemci ve 4 GB belleğe sahip olan bilgisayar üzerinde elde edilmiştir. EG ve EF ayrıklaştırma yöntemleri için aralık (k) sayısı 3 ile 25 arasında 2 artırarak ayarlanmıştır ve her aralıkta karar ağacı ID3, NB ve Karar tablo (KT) kural tabanlı sınıflandırma algoritmaları üzerinde sınıflandırma başarıları elde edilmiştir weka programını kulanarak ve weka programının ekran görüntüleri Şekil 4.3 ve 4.4'te verilmiştir. Aşağıda Çizelge 4.1, 4.2, 4.3 ve 4.4'te 3 ile 25 arasında 2 artırarak EG için sınıflandırma algoritmaları üzerinde sınıflandırma başarıları verilmektedir.

Çizelge 4.1 EG aralık 3, 5 ve 7 için sınıflandırma başarıları

Adı	Aralık sayısı = 3			Aralık sayısı = 5			Aralık sayısı = 7		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	78.72	**76.64**	**78.88**	**91.68**	66.08	40.16	91.68	66.08	40.16
Banana	61.26	61.96	61.98	62.83	68.47	68.47	64.08	64.83	65.77
Ecoli	80.36	72.02	**79.17**	**82.74**	**73.21**	65.77	81.55	71.13	61.61
Glass	48.13	44.86	40.65	48.13	48.13	45.79	58.88	51.87	44.86
Ionosphere	86.32	**84.62**	**79.77**	86.32	79.77	76.64	86.61	83.19	76.07
Iris	94.67	**95.33**	**96.67**	92.67	94.00	90.67	88.00	88.00	92.67
Magic	79.77	80.45	81.06	82.02	**82.00**	**81.75**	81.75	80.85	81.60
New thyroid	88.84	86.05	88.84	90.23	90.70	89.30	93.02	87.44	**92.09**
Pima	72.14	73.18	**72.40**	74.87	74.35	62.89	75.91	73.18	61.46
Ring	92.35	**78.72**	**87.51**	94.78	75.45	79.86	96.18	75.82	76.86
Image Segment	70.48	63.59	71.08	81.52	70.95	**79.48**	84.33	70.22	76.41
Wine	89.33	82.02	**88.20**	93.26	80.90	79.78	96.07	**86.52**	82.58
Yeast	46.16	43.46	**42.72**	49.39	41.31	39.08	53.30	44.20	38.34

Çizelge 4.2 EG aralık 9, 11 ve 13 için sınıflandırma başarıları

Adı	Aralık sayısı = 9			Aralık sayısı = 11			Aralık sayısı = 13		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	91.68	66.08	40.16	91.68	66.08	40.16	91.68	66.08	40.16
Banana	63.43	67.64	67.74	63.04	69.32	69.43	64.28	70.79	71.00
Ecoli	82.44	66.07	57.14	80.06	63.39	55.06	80.65	58.63	60.42
Glass	62.15	53.74	**50.93**	**65.42**	53.27	44.86	59.81	**58.88**	43.46
Ionosphere	86.32	84.05	77.49	86.61	81.77	70.37	87.18	79.77	70.09
Iris	93.33	94.67	90.67	94.67	93.33	88.00	**95.33**	93.33	91.33
Magic	82.93	81.07	81.09	82.43	80.31	79.34	82.69	80.56	77.94
New thyroid	93.02	**91.63**	89.30	92.56	91.16	88.84	93.02	87.91	91.63
Pima	75.26	73.57	57.16	75.26	74.09	61.33	75.52	74.09	59.90
Ring	96.96	74.59	75.08	97.18	75.32	73.15	97.45	74.36	69.97
Image Segment	84.59	66.58	76.84	84.24	**73.77**	75.41	87.84	71.56	74.55
Wine	94.94	79.78	75.28	96.63	79.78	71.91	95.51	81.46	71.91
Yeast	54.25	42.72	36.32	54.85	43.87	34.57	**55.59**	42.12	34.91

Çizelge4 .3 EG aralık 15, 17 ve 19 için sınıflandırma başarıları

Adı	Aralık sayısı = 15			Aralık sayısı = 17			Aralık sayısı = 19		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	91.68	66.08	40.16	91.68	66.08	40.16	91.68	66.08	40.16
Banana	65.64	**72.23**	72.58	66.51	70.94	71.19	66.68	71.11	72.11
Ecoli	79.17	57.14	52.68	77.98	61.01	55.36	78.87	60.42	50.89
Glass	60.75	57.48	47.20	57.48	54.67	44.86	57.01	49.53	44.39
Ionosphere	87.18	82.05	67.24	87.18	82.05	65.81	86.61	76.07	64.10
Iris	94.00	94.67	92.00	92.67	94.00	90.67	95.33	94.00	85.33
Magic	82.97	80.72	76.06	82.93	80.58	75.16	83.06	80.71	74.01
New thyroid	92.09	91.16	87.44	93.02	88.84	85.12	91.63	88.84	82.79
Pima	75.78	73.96	57.42	74.87	74.22	51.43	74.61	72.40	51.30
Ring	97.32	72.09	69.31	97.50	72.80	67.14	97.61	72.73	72.73
Image Segment	87.45	70.13	75.89	87.40	66.58	73.77	87.53	66.67	71.86
Wine	96.63	73.60	69.66	97.19	69.10	65.73	97.19	71.91	55.62
Yeast	55.73	41.44	30.26	54.85	40.84	30.59	54.11	**44.74**	28.03

Çizelge 4.4 EG aralık 21, 23 ve 25 için sınıflandırma başarıları

Adı	Aralık sayısı = 21			Aralık sayısı = 23			Aralık sayısı = 25		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	91.68	66.08	40.16	91.68	66.08	40.16	91.68	66.08	40.16
Banana	65.81	71.06	**72.92**	65.98	71.42	72.66	**66.91**	70.25	72.28
Ecoli	76.79	63.10	46.73	75.30	59.52	44.35	77.98	59.23	43.45
Glass	58.88	48.13	41.59	59.35	47.20	43.93	59.35	52.80	42.06
Ionosphere	87.18	81.20	65.24	**87.46**	82.05	65.53	86.61	82.34	64.67
Iris	94.00	93.33	88.00	93.33	92.67	84.00	94.67	92.67	81.33
Magic	83.13	80.53	73.00	**83.21**	80.73	72.42	82.84	80.11	70.84
New thyroid	92.09	88.37	82.33	91.63	86.51	78.14	**93.49**	85.58	81.86
Pima	76.30	**74.74**	51.30	**76.95**	71.74	54.04	75.39	73.57	48.70
Ring	97.65	72.36	62.81	**97.69**	72.05	62.08	97.64	71.15	61.22
Image Segment	87.45	65.54	70.22	86.80	69.39	71.17	**88.83**	73.25	69.44
Wine	**97.75**	72.47	66.29	96.07	75.28	62.36	96.63	63.48	58.99
Yeast	54.38	39.76	26.82	52.70	40.84	26.28	53.37	39.56	27.43

EG ayrıklaştırma yönteminin sınıflandırma algoritmaları üzerinde sınıflandırma başarılarına bakıldığı zaman, EG ayrıklaştırma yöntemi NB sınıflandırma algoritması için balance veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 91.68, banana veri stinin en iyi aralık k sayısı 25'te aralık sayısında başarısı 66.91, ecoli veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 82.74, glass veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 65.42, ionosphere veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 87.46, iris veri stinin en iyi aralık k sayısı 13'te aralık sayısında başarısı 95.33, magic veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 83.21, new thyroid veri stinin en iyi aralık k sayısı 25'te aralık sayısında başarısı 93.49, pima veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 76.95, ring veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 79.69, image segmentation veri stinin en iyi aralık k sayısı 25'te aralık sayısında başarısı 88.83, wine veri stinin en

iyi aralık k sayısı 21'de aralık sayısında başarısı 97.75, yeast veri stinin en iyi aralık k sayısı 13'te aralık sayısında başarısı 55.59, bo sonuclar elde edilmiştir.

KT sınıflandırma algoritması için balance veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 76.64, banana veri stinin en iyi aralık k sayısı 15'te aralık sayısında başarısı 72.23, ecoli veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 73.21, glass veri stinin en iyi aralık k sayısı 13'te aralık sayısında başarısı 58.88, ionosphere veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 84.62, iris veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 95.33, magic veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 82.00, new thyroid veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 91.63, pima veri stinin en iyi aralık k sayısı 21'de aralık sayısında başarısı 74.74, ring veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 78.72, image segmentation veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 73.77, wine veri stinin en iyi aralık k sayısı 7'de aralık sayısında başarısı 86.52, yeast veri stinin en iyi aralık k sayısı 19'da aralık sayısında başarısı 44.74, bo sonuclar elde edilmiştir.

ID3 sınıflandırma algoritması için balance veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 78.88, banana veri stinin en iyi aralık k sayısı 21'de aralık sayısında başarısı 72.92, ecoli veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 79.17, glass veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 50.93, ionosphere veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 79.77, iris veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 96.67, magic veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 81.75, new thyroid veri stinin en iyi aralık k sayısı 7'te aralık sayısında başarısı 92.09, pima veri stinin en iyi aralık k sayısı 3'de aralık sayısında başarısı 72.40, ring veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı87.51, image segmentation veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 79.48, wine veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 88.20, yeast veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 42.72, bo sonuclar elde edilmiştir.

EF ayrıklaştırma yöntemi için ise aşağıda sırasıyla Çizelge 4.5, 4.6, 4.7 ve 4.8'de sınıflandırma algoritmaları için sınıflandırma başarıları verilmektedir.

Çizelge 4.5 EF Aralık 3, 5 ve 7 için sınıflandırma başarıları

Adı	Aralık sayısı = 3			Aralık sayısı = 5			Aralık sayısı = 7		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	64.96	64.96	64.96	69.76	69.76	69.76	69.76	69.76	69.76
Banana	57.77	62.47	62.47	58.91	59.57	59.57	56.64	57.36	57.36
Ecoli	**69.94**	**67.86**	68.45	64.29	62.50	62.80	65.48	61.90	58.04
Glass	47.20	40.65	41.12	55.14	50.00	44.86	53.27	51.40	40.19
Ionosphere	83.76	83.48	76.92	86.04	84.33	80.63	89.17	89.74	82.91
Iris	61.33	62.00	64.67	64.00	63.33	62.67	67.33	65.33	68.00
Magic	73.83	74.97	72.80	78.27	79.64	77.97	79.47	80.56	77.01
New thyroid	87.91	83.72	84.65	92.09	88.37	89.77	90.23	89.77	86.51
Pima	73.70	72.01	66.67	73.70	73.05	62.76	71.74	70.31	64.71
Ring	73.01	67.42	61.58	85.77	75.34	71.41	82.73	72.28	66.28
Image Segment	60.65	54.89	58.83	53.29	46.62	52.81	72.86	62.47	73.20
Wine	76.40	55.06	67.42	76.97	63.48	**73.03**	75.84	69.66	65.17
Yeast	41.51	39.69	39.42	46.77	44.34	41.64	44.61	44.41	42.45

Çizelge 4.6 EF Aralık 9, 11 ve 13 için sınıflandırma başarıları

Adı	Aralık sayısı = 9			Aralık sayısı = 11			Aralık sayısı = 13		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	63.52	62.24	64.16	69.28	63.52	69.28	69.12	67.52	69.12
Banana	63.30	64.08	63.70	55.70	55.62	55.60	56.57	59.83	60.00
Ecoli	65.18	63.10	61.01	69.35	66.67	**85.71**	68.15	62.80	63.39
Glass	51.87	47.20	50.93	**61.68**	54.21	**95.79**	57.48	54.21	40.65
Ionosphere	**90.31**	90.60	**100.00**	88.89	**90.88**	100.00	89.17	88.89	80.63
Iris	66.67	64.00	66.67	74.67	74.00	**75.33**	72.00	72.00	72.00
Magic	79.15	80.94	85.83	79.51	**81.99**	**88.83**	**80.74**	81.17	76.56
New thyroid	**94.88**	**94.42**	**98.14**	92.09	92.09	97.21	92.56	91.16	89.77
Pima	**75.13**	**75.39**	91.15	74.35	73.96	**93.75**	74.61	73.83	60.55
Ring	82.66	72.92	99.96	85.43	76.00	**100.00**	**89.85**	75.77	68.74
Image Segment	76.32	**70.91**	**99.91**	**81.93**	68.83	97.23	77.36	69.44	76.02
Wine	80.90	60.67	70.22	75.28	60.67	63.48	79.21	55.06	58.99
Yeast	48.32	**48.11**	**61.46**	44.74	44.88	57.48	43.73	43.40	36.59

Çizelge 4.7 EF Aralık 15, 17 ve 19 için sınıflandırma başarıları

Adı	Aralık sayısı = 15			Aralık sayısı = 17			Aralık sayısı = 19		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	72.64	72.64	72.64	69.60	67.52	69.60	69.12	67.68	69.12
Banana	59.28	**64.19**	64.23	**64.98**	64.83	**64.83**	54.98	56.23	56.23
Ecoli	66.96	61.61	62.50	61.31	59.23	62.50	65.48	59.52	57.74
Glass	54.67	51.87	43.46	55.61	**54.67**	37.38	56.54	47.20	40.65
Ionosphere	88.60	85.47	81.77	87.18	85.76	74.64	88.89	86.04	83.76
Iris	68.67	68.00	68.00	68.67	67.33	65.33	66.67	65.33	67.33
Magic	78.21	81.01	78.07	79.14	81.62	75.57	79.38	81.09	76.36
New thyroid	92.56	90.23	84.65	93.02	88.84	87.91	93.02	86.51	86.05
Pima	74.48	73.96	60.03	74.09	73.44	67.97	74.48	72.79	62.50
Ring	87.51	73.08	66.82	87.95	**77.36**	71.84	83.64	68.50	62.73
Image Segment	72.99	67.32	69.31	79.26	70.35	75.24	80.13	67.66	77.66
Wine	74.16	55.06	53.93	78.09	67.42	64.61	71.91	50.00	47.75
Yeast	**48.45**	44.68	38.14	45.62	42.59	37.67	45.82	43.19	30.46

Çizelge 4.8 EF Aralık 21, 23 ve 25 için sınıflandırma başarıları

Adı	Aralık sayısı = 21			Aralık sayısı = 23			Aralık sayısı = 25		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	69.12	67.52	69.12	67.52	67.52	67.52	**77.60**	**77.60**	**77.60**
Banana	59.36	63.25	63.23	60.81	60.92	61.00	62.70	63.02	62.96
Ecoli	64.29	61.01	64.88	69.05	61.01	61.61	67.56	63.39	60.12
Glass	52.34	51.40	41.12	53.27	53.27	34.11	53.74	54.67	38.32
Ionosphere	88.60	86.61	82.34	87.46	87.75	81.77	88.60	88.03	79.77
Iris	66.67	66.67	66.00	**92.00**	**92.67**	88.67	92.00	88.67	90.00
Magic	79.90	80.80	78.06	78.72	80.53	77.12	79.73	81.37	76.97
New thyroid	94.42	92.56	88.84	92.56	89.30	84.65	93.02	88.37	86.98
Pima	75.00	74.22	59.51	73.57	73.70	61.85	73.31	73.70	63.80
Ring	87.72	74.68	69.92	84.23	69.99	61.78	89.14	72.66	66.65
Image Segment	75.37	67.88	74.59	79.48	69.09	73.64	78.48	66.88	74.72
Wine	**88.20**	70.22	72.47	84.83	**75.28**	69.10	83.71	81.46	70.79
Yeast	45.82	44.61	31.06	48.38	45.96	43.33	45.35	44.95	42.99

EF ayrıklaştırma yönteminin sınıflandırma algoritmaları üzerinde sınıflandırma başarılarına bakıldığı zaman, EF ayrıklaştırma yöntemi NB sınıflandırma algoritması için balance veri stinin en iyi aralık k sayısı 25'te aralık sayısında başarısı 77.60, banana veri stinin en iyi aralık k sayısı 17'de aralık sayısında başarısı 64.98, ecoli veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 69.74, glass veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 61.68, ionosphere veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 90.31, iris veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 9200, magic veri stinin en iyi aralık k sayısı 13'te aralık sayısında başarısı 80.74, new thyroid veri stinin en iyi aralık k sayısı 9'daaralık sayısında başarısı 94.88, pima veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 75.13, ring veri stinin en iyi aralık k sayısı 13'te aralık sayısında başarısı 89.85, image segmentation veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 81.93, wine veri stinin en iyi aralık k sayısı 21'de aralık sayısında başarısı 88.20, yeast veri stinin en iyi aralık k sayısı 15'te aralık sayısında başarısı 48.45, bo sonuclar elde edilmiştir.

KT sınıflandırma algoritması için balance veri stinin en iyi aralık k sayısı 25'te aralık sayısında başarısı 77.60, banana veri stinin en iyi aralık k sayısı 15'te aralık sayısında başarısı 64.19, ecoli veri stinin en iyi aralık k sayısı 3'te aralık sayısında başarısı 67.86, glass veri stinin en iyi aralık k sayısı 17'de aralık sayısında başarısı 54.67, ionosphere veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 90.88, iris veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 92.62, magic veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 81.99, new thyroid veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 94.42, pima veri stinin en iyi aralık k

sayısı 9'da aralık sayısında başarısı 75.39, ring veri stinin en iyi aralık k sayısı 17'de aralık sayısında başarısı 77.36, image segmentation veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 70.91, wine veri stinin en iyi aralık k sayısı 23'te aralık sayısında başarısı 75.28, yeast veri stinin en iyi aralık k sayısı 15'te aralık sayısında başarısı 48.45, bo sonuclar elde edilmiştir.

ID3 sınıflandırma algoritması için balance veri stinin en iyi aralık k sayısı 25'te aralık sayısında başarısı 77.60, banana veri stinin en iyi aralık k sayısı 17'de aralık sayısında başarısı 64.83, ecoli veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 85.71, glass veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 95.79, ionosphere veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 100.00, iris veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 75.33, magic veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 88.83, new thyroid veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 98.14, pima veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 93.75, ring veri stinin en iyi aralık k sayısı 11'de aralık sayısında başarısı 100.00, image segmentation veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 99.91, wine veri stinin en iyi aralık k sayısı 5'te aralık sayısında başarısı 73.03, yeast veri stinin en iyi aralık k sayısı 9'da aralık sayısında başarısı 61.46, bo sonuclar elde edilmiştir.

EG ve EF ayrıklaştırma yöntemlerinin en büyük sıkıntısı ise belirli bir veri seti için aralık k sayısının belli olmaması ve en iyi sınıflandırma başarısını gösteren aralık k sayısını belirleme problemidir. Yukarıda veri setlerinin testlerinde de görüldüğü gibi en iyi aralık k sayısını bulmak için belirli aralıklar denenerek en iyi sınıflandırma başarıları bulunmuştur. En iyi sınıflandırma başarısını gösteren aralık k sayılarının ortalaması alınarak sınıflandırma algoritmalarının sınıflandırma başarıları hesaplanmıştır. ID3 ayrıklaştırma yöntemi aralık k sayısının kullanıcı tarafından verilmesini ortadan kaldırarak mevcut bilgilerin bilgi kazancılarını hesaplayarak iyi aralık sayılarını elde eder, fakat ne yazık ki elde edilen aralık k sayısı ayrıklaştırılan veriler için en iyi aralık k sayısı değildir. ID3 ayrıklaştırma yöntemi, aralık k sayısını verilerin içinden sınıf bilgilerine göre elde eder. EG ve EF ayrıklaştırma yöntemlerinin sonuçları ID3 ayrıklaştırma yöntemi ile karşılaştırılmıştır. Bu karşılaştırmada EG ve EF ayrıklaştırma yöntemlerinin en iyi sınıflandırma başarısını elde eden aralık k sayılarının ortalaması alınarak tüm veri setleri için bir ortak ortalama aralık k sayısı bulunmaktadır. EG ayrıklaştırma yönteminde ortalama aralık k sayısı 3 ve EF ayrıklaştırma yönteminde ise 9 olarak bulunmuştur. Aşağıda Çizelge 4.9'da en iyi aralık k sayısına göre EG, EF ve

ID3 ayrıklaştırma yöntemleri için sınıflandırma algoritmalarının sınıflandırma başarıları gösterilmiştir.

Çizelge 4.9 EG, EF ve ID3 ayrıklaştırma yöntemlerinin sınıflandırma başarı sonuçları

Adı	EG Aralık sayısı = 3			EF Aralık sayısı = 9			ID3		
	NB	KT	ID3	NB	KT	ID3	NB	KT	ID3
Balance	78.72	**76.64**	**78.88**	63.52	62.24	64.16	71.52	**72.16**	**69.44**
Banana	61.26	61.96	61.98	63.30	64.08	63.70	**73.74**	**75.55**	**75.57**
Ecoli	80.36	72.02	**79.17**	65.18	63.10	61.01	**86.31**	**76.79**	**79.46**
Glass	48.13	44.86	40.65	51.87	47.20	50.93	**71.50**	**69.16**	**75.70**
Ionosphere	86.32	**84.62**	**79.77**	**90.31**	90.60	**100.00**	**90.31**	**90.31**	**88.60**
Iris	94.67	**95.33**	**96.67**	66.67	64.00	66.67	**94.67**	**94.00**	**94.00**
Magic	79.77	80.45	81.06	79.15	80.94	85.83	78.30	**82.82**	79.01
New thyroid	88.84	86.05	88.84	**94.88**	**94.42**	**98.14**	**96.28**	**95.35**	**95.35**
Pima	72.14	73.18	**72.40**	**75.13**	**75.39**	91.15	**78.13**	**79.04**	**77.21**
Ring	92.35	**78.72**	**87.51**	82.66	72.92	99.96	**97.93**	**76.07**	**76.35**
Image Segment	70.48	63.59	71.08	76.32	**70.91**	**99.91**	**91.65**	**90.39**	**94.89**
Wine	89.33	82.02	**88.20**	80.90	60.67	70.22	**98.88**	**92.70**	**96.07**
Yeast	46.16	43.46	**42.72**	48.32	**48.11**	**61.46**	**59.37**	**57.82**	**58.69**

Çizelge 4.9'da sonuçlarını ele alındığında, EG ayrıklaştırma yöntemi KT sınıflandırma algoritmasında sırasıyla balance76.64 , Ionosphere84.62 , iris95.33 ve Ring78.72 veri setlerinde sınıflandırma başarıları elde etmiştir. ID3 sınıflandırma algoritmasında ise sırasıyla Balance78.88, Ecoli79.17, Ionosphere79.77, Iris96.67, Pima72.40, Ring87.51, Wine88.20, Yeast42.72 veri setlerinde iyi bir sonuç vermiştir. ID3 ayrıklaştırma yöntemi ise NB sınıflandırma algoritmasında banana73.74, ecoli86.31, glass71.50, ionosphere90.31, iris94.67, newthyroid96.28, pima78.13, ring97.93, image segmentation91.65, wine98.88, ve yeast59.37 veri setleri için sınıflandırma başarıları elde edilmiştir. KT sınıflandırma algoritması için ise balance72.16, banana75.55, ecoli76.79, glass69.16, ionosphere90.31, iris94.00, magic82.82, newthyroid95.35, pima79.04, ring76.07, image segmentation90.39, wine92.70, ve yeast57.82 setleri için sınıflandırma başarıları elde edilmiştir. ID3 sınıflandırma algoritması için ise balance69.44, banana75.57, ecoli79.46, glass75.70, ionosphere88.60, iris94.00, newthyroid95.35, pima77.21, ring76.35, image segmentation94.89, wine96.07, ve yeast58.69 setleri için sınıflandırma başarıları elde edilmiştir. Genel olarak ayrıklaştırma yöntemlerinin sınıflandırma algoritmaları üzerinde test sonuçlarına baktığımız zaman ID3 ayrıklaştırma yöntemi EW ve EF ayrıklaştırma yöntemlerinden veri setleri üzerinde daha iyi bir performans sergilemiştir. EG, EF ve ID3 ayrıklaştırma işlemlerine tabi tutulan veri kümelerinin NB ve ID3 sınıflandırma başarıları literatürde araştımacılar tarafından önerilen ayrıklaştırma

algoritmaları ile karşılaştırılmıştır. Literatürde proportional discretization (PD), fixed frequency discretization (FFD), Lazy discretization (LD) ve entropy minimization discretization (EMD) ayrıklaştırma algoritmaları ile karşılaştırılmıştır (Yang ve Webb, 2009).Çizelge 4.10'da veri kümeleri NB sınıflandırma algoritmasının başarıları verilmiştir.

Çizelge 4.10 EW, EF, ID3 ve diğer ayrıklaştırma yöntemlerinin NB sınıflandırma başarının sonuçları

Adı	EMD	LD	PD	FFD	EG	EF	ID3
Glass	65.1	68	67.4	60.9	48.13	51.87	**71.50**
Ionosphere	88.9	89.2	89.6	89.3	86.32	**90.31**	**90.31**
Iris	93.2	93.3	93.6	92.9	**94.67**	66.67	**94.67**
New thyroid	**98.3**	97.6	98.2	98.2	88.84	94.88	96.28
Pima	74	74.6	74	73.5	72.14	75.13	**78.13**
Wine	97.4	97.1	97.6	97.2	89.33	80.90	**98.88**

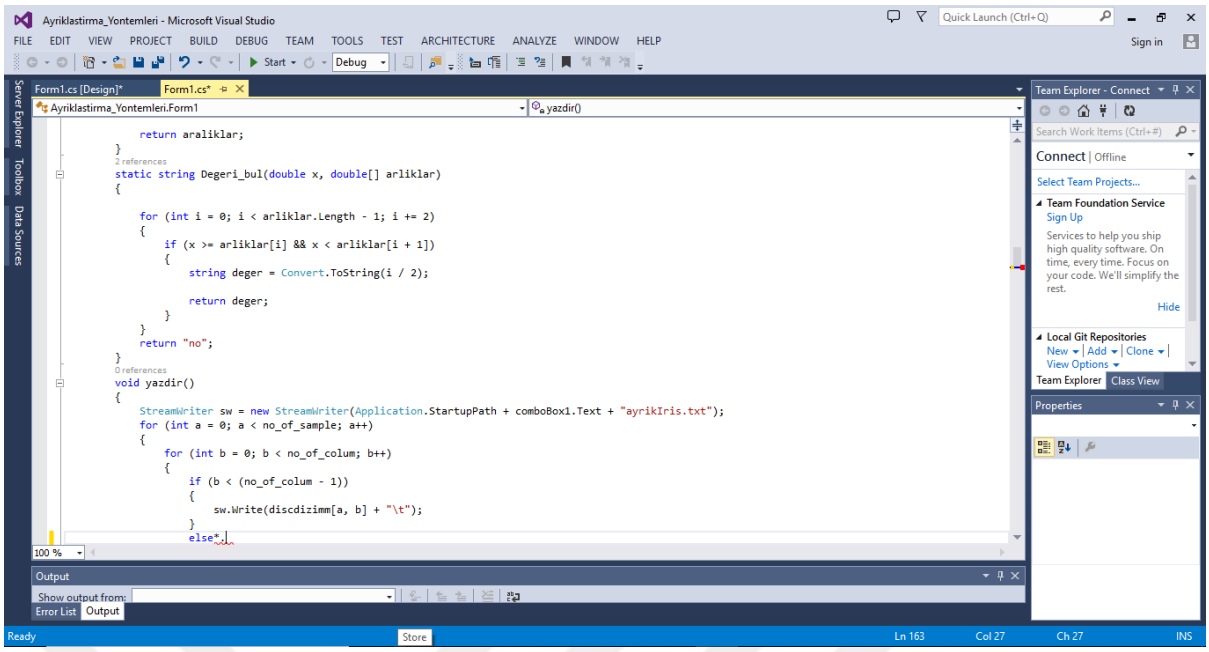
Çizelge 4.10'da NB sınıflandırma başarılarına bakıldığında, Glass, Lonosphere, Iris, Pima ve Wine veri kümelerinde Entropi tabanlı ID3 ayrıklaştırma algoritması literatürdeki ayrıklaştırma yöntemlerinden daha iyi sonuç vermiştir.

EG, EF ve ID3 ayrıklaştırma işlemlerine tabi tutulan veri kümelerinin NB ve ID3 sınıflandırma başarıları literatürde araştımacılar tarafından önerilen ayrıklaştırma algoritmaları ile karşılaştırılmıştır. Literatürde class-attribute dependent discretization, or simply CADD, class-attribute interdependency maximization (CAIM), class-attribute contingency coefficient algorithm (CACC) ve A novel class-attribute inter dependency discretization algorithm (NCAIC) ayrıklaştırma algoritmaları ile karşılaştırılmıştır (Yan ve ark., 2014a). Çizelge 4.11'de veri kümeleri NB sınıflandırma algoritmasının başarıları verilmiştir.

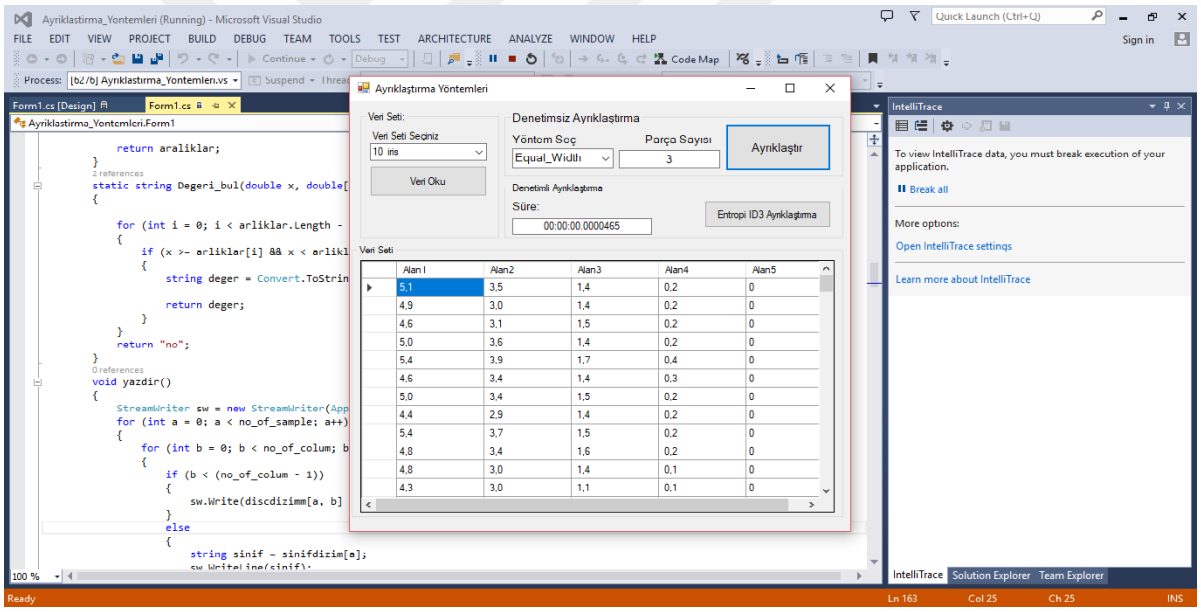
Çizelge 4.11 EW, EF, ID3 ve diğer ayrıklaştırma yöntemlerinin ID3 sınıflandırma başarının sonuçları

Adı	CADD	CAIM	CACC	NCAIC	MDL	EG	EF	ID3
Glass	72.1	67.4	67.4	67.4	58.1	40.65	50.93	**75.70**
Ionosphere	86.9	85.9	83.1	88.7	84.5	79.77	**100.00**	88.60
Iris	90	90	93.3	**96.7**	90	96.67	66.67	94.00
Pima	63.6	63.6	76	76	67.5	72.40	**91.15**	77.21
Wine	91.7	91.7	86.1	94.4	83.3	88.20	70.22	**96.07**

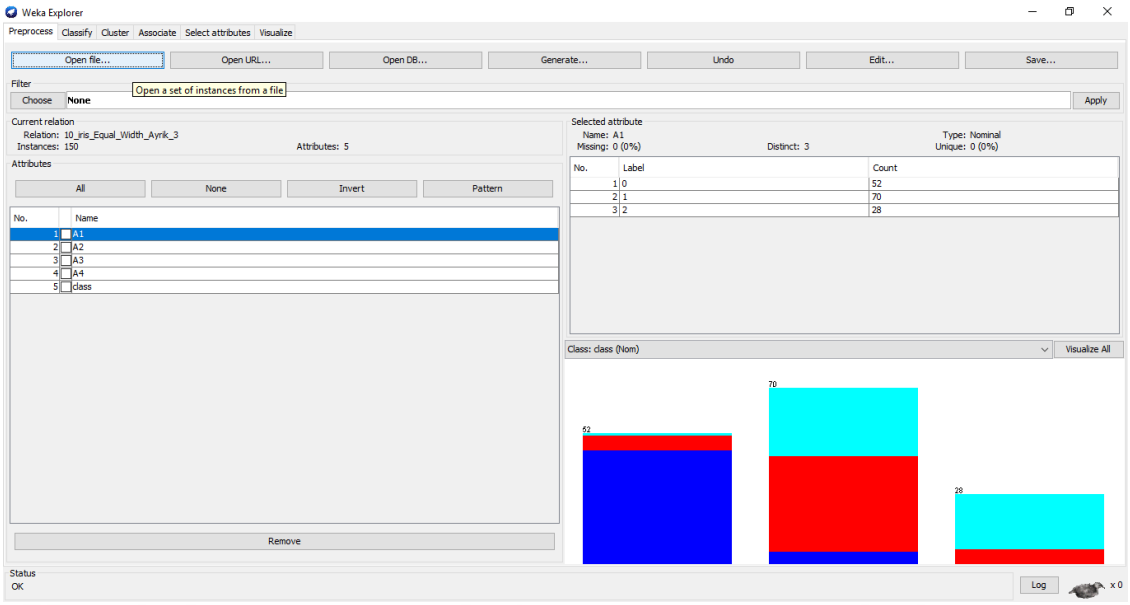
Çizelge 4.11'de ID3 sınıflandırma başarılarına bakıldığında, Glass ve Wine veri setleri ID3 ayrıklaştırma algoritmasında diğer algoritmalardan iyi sonuç vermiştir. Lonosphere, Iris ve Pima veri setlerinde sırasıyla EF, NCAIC ve EF ayrıklaştırma algoritmaları diğer algoritmalardan daha iyi bir sınıflandırma başarısı vermiştir.



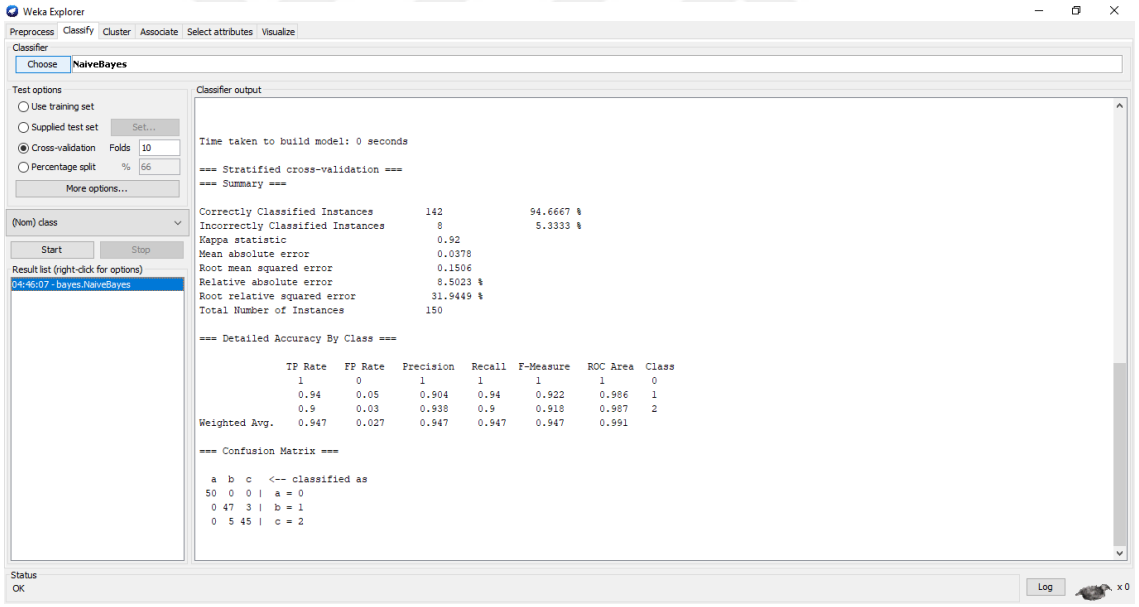
Şekil 4.1 İlk ekran görüntüsü (visual studio)



Şekil 4.2 Uyguladığımızdaki ekran görüntüsü (visual studio)



Şekil 4.3 weka programının ilk ekran görüntüsü



Şekil 4.4 weka prgramının sınıflandırma sonucunun ekran görüntüsü

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Veri ön işleme, VM veya veri analizinde ilk ve kritik adımlardan biridir. Veri ön işleme VM algoritmalarının performanslarını doğrudan etkilemektedir. İyi bir veri ön işleme yalnızca VM algoritmalarının doğruluğunu artırmakla kalmaz, aynı zamanda algoritmanın verimliliğini de çarpıcı şekilde artırır. Genel olarak, veri ön işleme, VM algoritmasının uygulanmasından önce işlenen veri temizliği, veri entegrasyonu, veri geçişi, veri azaltma ve veri ayrıklaştırma anlamına gelmektedir. VM algoritmaları, matematik, bilgisayar ve istatistik vb. farklı uygulama etki alanlarında kullanmadan önce, verilerin önceden işlenmesi için çeşitli işlevlere ihtiyaç duymaktadır. VM alanlarında çalışan araştırmacılar tarafından çeşitli veri ön işleme teknikleri önerilmiştir. Sınıflandırma, VM algoritmaları arasında en temel algoritmadır ve tıbbi teşhis, karar analizi, makine öğrenmesi, bilgi alma ve havacılık alanlarında başarılı bir şekilde kullanılmaktadır. Oysa sınıflandırma tekniklerinin çoğu pratikte tatmin edici sonuçlar elde edemez, asıl sebep, özellikle orijinal verilerin hızla artması ile birlikte, bol miktarda verinin etkilenmesi nedeniyle işlem sırasında düşük performansa sahip olmalarıdır. Aynı zamanda bazı sınıflandırma algoritmaları sürekli veri setleri üzerinde çok iyi bir sınıflandırma performansları sergilemez. Bu problemin üstesinden gelmek için sınıflandırma algoritmaları için ayrıklaştırma veri ön işleme teknikleri geliştirilmiştir. Bu tez çalışmasında sınıflandırma algoritmaları için geliştirilen ayrıklaştırma yöntemleri çok ayrıntılı olarak anlatılmıştır ve sınıflandırma başarısına olan etkileri incelenmiştir. EG, EF ve ID3 ayrıklaştırma yöntemleri sık kullanılan ayrıklaştırma yöntemleridir. Bu yöntemlerin sınıflandırma algoritmasına olan etkisi UCI veri setinde Balance, Banana, Ecoli, Glass, Ionosphere, Iris, Magic, New thyroid, Pima, Ring, Image Segment, Wine ve Yeast veri setlerini sınıflandırma için kullanılmıştır. NB, KT ve ID3 sınıflandırma algoritmalarında daha iyi sınıflandırma performansı elde etmesi için veri setleri ayrıklaştırma işlemine tabi tutulmuştur. Bu veri setleri ilk önce k 3 ve 25 değerleri arasında 2 artırarak EG ve EF ayrıklaştırma yöntemlerine uygulandıktan sonra NB, KT ve ID3 sınıflandırma algoritmaları sınıflandırmalarına verilmiştir ve sınıflandırma sonuçları elde edilmiştir. Daha sonra aynı veri setleri ID3 danışmanlı ayrıklaştırma yöntemi ile ayrıklaştırıldıktan sonra NB, KT ve ID3 sınıflandırma algoritmaları ile sınıflandırılmıştır. NB, KT ve ID3 sınıflandırma algoritmalarından elde edilen iki farklı sınıflandırma başarıları karşılaştırılmıştır. Bu

karşılaştırma sonucuna göre, EG ve EF ayrıklaştırma yöntemleri NB, KT ve ID3 sınıflandırma algoritmaları için ID3 ayrıklaştırma yöntemine göre iyi bir sınıflandırma başarısı sergilememiştir. Bunun temel sebebi ise EG ve EF ayrıklaştırma yöntemlerinde verilerin aralık sayılarının kullanıcı tarafından verilmesidir. ID3 ayrıklaştırma yönteminde ise verilerin aralık sayıları verilerin bilgi kazancına göre elde edilmektedir. Sonuç olarak, ID3 ayrıklaştırma yöntemi EG ve EF ayrıklaştırma yöntemlerinden daha iyi bir şekilde NB, KT ve ID3 sınıflandırma algoritmalarının sınıflandırma performanslarını artırmıştır.

5.2 Öneriler

VM, birçok gerçek dünya problemleri üzerinde veri analizi için sıklıkla kullanılmaktadır. Bu problemlerin verilerinin iyi bir şekilde VM ile analiz edilebilmesi için, bu verilerin VM algoritmasına uygun bir veri önişleme tekniği ile işlenmesi gerekmektedir. Veri önişleme, VM algoritmaları için çok önemli bir işlemdir ve normalizasyon, veri azaltma ve ayrıklaştırma gibi işlemleri kapsamaktadır. Genelde birçok sınıflandırma algoritmaları ayrık veri setleri üzerinde daha iyi bir sınıflandırma başarısı sergilemektedir. Dolayısıyla ayrıklaştırma işlemi birçok sınıflandırma algoritması için bir veri önişleme tekniği olarak kullanılmaktadır. Ayrıklaştırma işlemi genel anlamıyla sayısal veri setlerini kategori veri setlerine dönüştürmektedir. Bu dönüştürme sonucunda veriler daha basit ve daha az alan kaplamaktadır. Bu avantajların yanı sıra ayrık tabanlı sınıflandırma algoritmalarının sayısal veri setleri üzerinde sınıflandırma başarısını artırmaktadır. Bu tez çalışmasında, sıklıkla kullanılan EG, EF ve Entropi ID3 ayrıklaştırma yöntemleri UCI 13 veri setine uygulandıktan sonra ID3, NB ve KT sınıflandırma algoritmaları ile sınıflandırılmıştır. Deney sonuçlarına göre, sayısal veri setlerine ayrıklaştırma yöntemleri uygulandıktan sonra daha iyi bir sınıflandırma başarısı elde edilmektedir. Bu tez çalışmasında araştırmacılar tarafından önerilen EG, EF ve Entropi ID3 ayrıklaştırma yöntemlerinin hangi sınıflandırma algoritmasında daha iyi sonuç verdiği incelenmiştir. Buna ek olarak karar ağaçları ve bayes sınıflandırıcılarında daha iyi bir sınıflandırma başarısı için tıbbi, market ve finans veri setleri için ID3 ve EF ayrıklaştırma teknikleri önerilebilir. Aynı zamanda veri setlerinin hafıza birimlerinde daha az yer kaplaması için ayrıklaştırma yöntemleri önerilebilir.

KAYNAKLAR

Aeberhard, S., 1991, UCI machine learning repository http://archive.ics.uci.edu/ml/datasets/Wine:

Bakar, A. A., Othman, Z. A. ve Shuib, N. L. M., 2009, Building a new taxonomy for data discretization techniques, *Data Mining and Optimization*, 132-140.

Boulle, M., 2004, Khiops: A statistical discretization method of continuous attributes, *Machine learning*, 55 (1), 53-69.

Boulle, M., 2005, Optimal bin number for equal frequency discretizations in supervized learning, *Intelligent Data Analysis*, 9 (2), 175-188.

Boullé, M., 2006, MODL: a Bayes optimal discretization method for continuous attributes, *Machine learning*, 65 (1), 131-165.

Chlebus, B. S. ve Nguyen, S. H., 1998, On finding optimal discretizations for two attributes, *International Conference on Rough Sets and Current Trends in Computing*, 537-544.

Chmielewski, M. R. ve Grzymala-Busse, J. W., 1994, Global discretization of continuous attributes as preprocessing for machine learning, *Third international workshop on rough sets and soft computing*, 294-301.

Cios, K. J., Swiniarski, R. W., Pedrycz, W. ve Kurgan, L. A., 2007, The knowledge discovery process, *Data Mining*, 9-24.

Creator ve Maintainer, 1996, UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/Yeast:

Dash, R. ve Paramguru, R. L., 2011, Comparative analysis of supervised and unsupervised discretization techniques, *International Journal of Advances in Science and Technology*, 2 (3), 29-37.

Dougherty, J., Kohavi, R. ve Sahami, M., 1995a, Supervised and unsupervised discretization of continuous features, *Machine Learning: Proceedings of the Twelfth İnternational Conference*, 194-202.

Dougherty, J., Kohavi, R. ve Sahami, M., 1995b, Supervised and unsupervised discretization of continuous features, In: Machine Learning Proceedings 1995, Eds: Elsevier, p. 194-202.

Draper , D., 1993, UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/Challenger+USA+Space+Shuttle+O-Ring:

Du, W. ve Zhan, Z., 2002, Building decision tree classifier on private data, 14.

Fayyad, U. ve Irani, K., 1993, Multi-interval discretization of continuous-valued attributes for classification learning.

Fisher, R. A., 1936, UCI machine learning repository, https://archive.ics.uci.edu/ml/datasets/iris:

Flores, M. J., Gámez, J. A., Martínez, A. M. ve Puerta, J. M., 2011, Handling numeric attributes when comparing Bayesian network classifiers: does the discretization method matter?, *Applied Intelligence*, 34 (3), 372-385.

Garcı´a, S., Luengo, J., Sa´ez, J. A., Lo´pez, V. ve Herrera, F., 2013, A survey of discretization techniques, *Transactions On Knowledge And Data Engineering*, 25.

Garcia, S., Luengo, J., Sáez, J. A., Lopez, V. ve Herrera, F., 2013, A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning, *IEEE Transactions on Knowledge and Data Engineering*, 25 (4), 734-750.

Group, V., 1990, UCI machine learning repository, http://archive.ics.uci.edu/ml/datasets/image+segmentation:

Gupta, A., Mehrotra, K. G. ve Mohan, C., 2010a, A clustering-based discretization for supervised learning, *Statistics & Probability Letters*, 80 (9), 816-824.
Gupta, A., Mehrotra, K. G. ve Mohan, C., 2010b, A clustering-based discretization for supervised learning, *Statistics & probability letters*, 80 (9-10), 816-824.
Hacibeyoglu, M., Arslan, A. ve Kahramanli, S., 2011, Improving Classification Accuracy with Discretization on Data Sets Including Continuous Valued Features, *Ionosphere*, 34 (351), 2.
Hacıbeyoğlu, M. ve Ibrahim, M. H., 2016, Comparison of the effect of unsupervised and supervised discretization methods on classification process, *Intelligent Systems and Applications in Engineering*, 4, 105–108.
Han, J., Pei, J. ve Kamber, M., 2006, Data Mining, *Elsevier Science*, Elsevier Science, p. p.
Han, J., Pei, J. ve Kamber, M., 2011, Data mining, *concepts and techniques*.
Horton, P., 1996, UCI machine learning repository

Hsu, C.-N., Huang, H.-J. ve Wong, T.-T., 2003, Implications of the Dirichlet assumption for discretization of continuous variables in naive Bayesian classifiers, *Machine learning*, 53 (3), 235-263.
Hume, T., 1994-04-22, UCI machine learning repository
http://archive.ics.uci.edu/ml/datasets/balance+scale:
Jiang, F. ve Sui, Y., 2015, A novel approach for discretization of continuous attributes in rough set theory, *Knowledge-Based Systems*, 73, 324-334.
Jiang, S.-y., Li, X., Zheng, Q. ve Wang, L.-x., 2009, Approximate equal frequency discretization method, *Intelligent Systems, 2009. GCIS'09. WRI Global Congress on*, 514-518.
Kaya, E., Findik, O., Babaoglu, I. ve Arslan, A., 2011, Effect of discretization method on the diagnosis of Parkinson's disease, *Int. J. Innov. Comp. Inf. Control*, 7, 4669-4678.
Khoo, L. P., Tor, S. B. ve Zhai, L. Y., 1999, A rough-set-based approach for classification
and rule induction, *Adv Manuf Technol*.
Kohavi, R. ve Sahami, M., 1996, Error-based and entropy-based discretization of continuous features, *KDD*, 114-119.
Kotsiantis, S. ve Kanellopoulos, D., 2006a, Discretization techniques: A recent survey, *GESTS International Transactions on Computer Science and Engineering*, 32 (1), 47-58.
Kotsiantis, S. ve Kanellopoulos, D., 2006b, Discretization techniques, *International Transactions on Computer Science and Engineering*, 32 (1), 47-58.
Li, R.-P. ve Wang, Z.-O., 2002, An entropy-based discretization method for classification rules with inconsistency checking, *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, 243-246.
Liu, L., Wong, A. K. ve Wang, Y., 2004, A global optimal algorithm for class-dependent discretization of continuous data, *Intelligent Data Analysis*, 8 (2), 151-170.
Madhu, G., Rajinikanth, T. ve Govardhan, A., 2014, Improve the classifier accuracy for continuous attributes in biomedical datasets using a new discretization method, *Procedia Computer Science*, 31, 671-679.
Majak, J., Shvartsman, B., Karjust, K., Mikola, M., Haavajõe, A. ve Pohlak, M., 2015, On the accuracy of the Haar wavelet discretization method, *Composites Part B: Engineering*, 80, 321-327.

Monti, S. ve Cooper, G. F., 1998, A multivariate discretization method for learning Bayesian networks from mixed data, *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 404-413.

Murphy, K. P., 2006, Naive bayes classifiers.

Peng, L., Qing, W. ve Yujia, G., 2009, Study on comparison of discretization methods, *Artificial Intelligence and Computational Intelligence*, 380-384.

Perner, P. ve Trautzsch, S., 1998, Multi-interval discretization methods for decision tree learning, *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 475-482.

Quinlan, R., 1987, UCI machine learning repository, http://archive.ics.uci.edu/ml/datasets/thyroid+disease:

Roy, A. ve Pal, S. K., 2003, Fuzzy discretization of feature space for a rough set classifier, *Pattern Recognition Letters*, 24 (6), 895-902.

Savicky, P., 2007, UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope:

Shen, L. ve Tay, F. E., 2001, A discretization method for rough sets theory, *Intelligent Data Analysis*, 5 (5), 431-438.

Sigillito, v., 1989, UCI machine learning repository https://archive.ics.uci.edu/ml/datasets/ionosphere:

Su, C.-T. ve Hsu, J.-H., 2005, An extended chi2 algorithm for discretization of real value attributes, *IEEE transactions on knowledge and data engineering*, 17 (3), 437-441.

Vina Spiehler, P. D., DABFT 1987, UCI machine learning repository, https://archive.ics.uci.edu/ml/datasets/glass+identification:

Witten, I. H., Frank, E., Hall, M. A. ve Pal, C. J., 2016a, Data Mining: Practical machine learning tools and techniques, *Morgan Kaufmann*, P.

Witten, I. H., Frank, E., Hall, M. A. ve Pal, C. J., 2016b, Data Mining, *Practical machine learning tools and techniques*.

Wu, X. ve Kumar, V., 2009, The top ten algorithm in data mining, *International Standard Book*, 13 (978-971).

Yan, D., Liu, D. ve Sang, Y., 2014a, A new approach for discretizing continuous attributes in learning systems, *Neurocomputing*, 133, 507-511.

Yan, D., Liu, D. ve Sang, Y., 2014b, A new approach for discretizing continuous attributes in learning systems, *Neurocomputing*, 133 (507-511).

Yang, Y. ve Webb, G. I., 2009, Discretization for naive-Bayes learning: managing discretization bias and variance, *Machine learning*, 74 (1), 39-74.

EKLER

EK-1 EG, EF ve ID3 ayrıklaştırma yöntemlerinin C# kodları

```
private void button1_Click(object sender, EventArgs e)
      {
         try
         {

            FileInfo f = new FileInfo(tablo);
            string fullname = f.FullName;
            fullname = "D:\\Datasets\\" + tablo;
            weka.core.Instances insts = new weka.core.Instances (new java.io.FileReader(
fullname + ".arff"));
            insts.setClassIndex(insts.numAttributes() - 1);
            Stopwatch dt = new Stopwatch();
            dt.Start();
            //EB
            weka.filters.supervised.attribute.Discretize ayrik = new weka.filters.supervise
d.attribute.Discretize();
            ayrik.setInputFormat(insts);
            weka.core.Instances ornekler;
            ornekler = weka.filters.Filter.useFilter(insts, ayrik);
            java.io.BufferedWriter writer = new java.io.BufferedWriter(new java.io.File
Writer("D:\\Ayrik_veriler\\" + tablo + "_Entropi.arff"));
            writer.write(ornekler.toString());
            writer.flush();
            writer.close();
            dt.Stop();
            textBox1.Text = Convert.ToString(dt.Elapsed);
            MessageBox.Show("işlem Tamam");
         }
         catch (java.lang.Exception ex)
         {
            ex.printStackTrace();
         }
      }


if (comboBox1.Text != "")
         {
            tablo_ayrik = combo_veriadi.Text;
            discdizimm = new string[no_of_sample, no_of_colum - 1];
            parca_sayisi = Convert.ToInt32(text_parcasayi.Text);
            string ayrik_yontem = comboBox1.Text;
            double[] dizisutunleri = new double[no_of_sample];
            double[] sirali_dizisutunleri = new double[no_of_sample];
            StreamWriter sw = new StreamWriter(Application.StartupPath + comboBox1.
Text + "_" + tablo_ayrik + "_" + ayrik_yontem + "_Aralık_" + parca_sayisi + ".txt");
```

```
            List<List<string>> parcalar_olmayan = new List<List<string>>();
            for (int kacsutunvasayac = 0; kacsutunvasayac < no_of_colum -
 1; kacsutunvasayac++)
            {
               parcalar_olmayan.Add(new List<string>());
               if (olmasin[kacsutunvasayac] != null)
               {

                  string[] olmayan_sutun = new string[no_of_sample];
                  for (int olmayan_sayac = 0; olmayan_sayac < no_of_sample; olmayan_s
ayac++)
                  {
                     discdizimm[olmayan_sayac, kacsutunvasayac] = ozelikdizim[olmayan
_sayac][kacsutunvasayac].ToString();
                     olmayan_sutun[olmayan_sayac] = discdizimm[olmayan_sayac, kacsut
unvasayac];
                     if (!(parcalar_olmayan[kacsutunvasayac].Contains(olmayan_sutun[ol
mayan_sayac]))) //olmayanlarda kaç parça var
                     {
                        parcalar_olmayan[kacsutunvasayac].Add(discdizimm[olmayan_say
ac, kacsutunvasayac]);
                     }
                  }
               }
               else
               {
                  for (int j = 0; j < ozelikdizim.Length; j++)
                  {
                     dizisutunleri[j] = Convert.ToDouble(ozelikdizim[j][kacsutunvasayac])
;
                  }
                  Array.Copy(dizisutunleri, sirali_dizisutunleri, ozelikdizim.Length);
                  Array.Sort(sirali_dizisutunleri);
                  switch (comboBox1.SelectedIndex)
                  {
                     case 0:
                        {
                           List<List<double>> aralikdegerler = new List<List<double>>();
                           for (int listdongu = 0; listdongu < parca_sayisi; listdongu++)
                           {
                              aralikdegerler.Add(new List<double>());
                           }
                           Stopwatch dt = new Stopwatch();
                           dt.Start();
                           double[] aralikli_dizi = arlik_olustur_Width(sirali_dizisutunleri,
parca_sayisi);
                           double toplama = 0;
                           for (int k = 0; k < dizisutunleri.Length; k++)
                           {
                              string discrete = Degeri_bul(dizisutunleri[k], aralikli_dizi);
```

```
                        discdizimm[k, kacsutunvasayac] = discrete;
                        aralikdegerler[Convert.ToInt32(discrete)].Add(dizisutunleri[k]
);
                    }
                    dt.Stop();
                    textBox1.Text = Convert.ToString(dt.Elapsed);
                    string[] somthing = new string[parca_sayisi];
                    for (int listdongu = 0; listdongu < parca_sayisi; listdongu++)
                    {
                        for (int ikin = 0; ikin < aralikdegerler[listdongu].Count; ikin++
)
                        {
                            toplama += aralikdegerler[listdongu][ikin];
                        }
                        somthing[listdongu] = (toplama / aralikdegerler[listdongu].Cou
nt).ToString("f3");
                    }
                    //yazdir();
                    int sayac = 0;
                    sw.WriteLine(kacsutunvasayac + 1 + " Nitelik için");
                    for (int aralik_sayac = 0; aralik_sayac < aralikli_dizi.Length; arali
k_sayac += 2)
                    {
                        sw.WriteLine(aralikli_dizi[aralik_sayac] + " <= " + sayac + ".
Aralık  " + " < " + aralikli_dizi[aralik_sayac + 1]);
                        sayac++;
                    }
                }
                break;
            case 1:
                {
                    Stopwatch dt = new Stopwatch();
                    dt.Start();
                    double[] aralikli_dizi = arlik_olustur_Freq(dizisutunleri, parca_sa
yisi);
                    for (int k = 0; k < dizisutunleri.Length; k++)
                    {
                        string discrete = Degeri_bul(dizisutunleri[k], aralikli_dizi);
                        discdizimm[k, kacsutunvasayac] = discrete;
                    }
                    dt.Stop();
                    textBox1.Text = Convert.ToString(dt.Elapsed);
                    //yazdir();
                    int sayac = 1;

                    sw.WriteLine(kacsutunvasayac + 1 + " Nitelik için");
                    for (int aralik_sayac = 0; aralik_sayac < aralikli_dizi.Length; arali
k_sayac += 2)
                    {
```

```
                        sw.WriteLine(aralikli_dizi[aralik_sayac] + " <= " + sayac + ".
Aralık  " + " < " + aralikli_dizi[aralik_sayac + 1]);
                        sayac++;
                    }
                }
                break;
            }
        }
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Sanarya Samal FAROOQ FAROOQ
Uyruğu : IRAK
Doğum Yeri ve Tarihi : KERKÜK 21/ Kasım /1993
Telefon : 05317357477
Faks :
E-Posta : Sanarya.nazdar@gmail.com

EĞİTİM

Derece		Adı	İlçe	İl	Bitirme Yılı
Lise	:	**AL-İzdihar Kız Lisesi,**	**Kerkük**		**2011**
Üniversite	:	**AL-Qalam Üniversitesi,**	**Kerkük**		**2015**
Yüksek Lisans	:				
Doktora	:				

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi

UZMANLIK ALANI

YABANCI DİLLER

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR

Selek, M. VE FAROOQ, S. 2018, Sürekli Veri Setlerinin Farklı Ayrıklaştırma Yöntemleri Kullanılarak Kural Tabanlı Sınıflandırma Algoritmalarına Uyarlanması ve Karşılaştırılması,2^{nd}International Symposium Humanities and Social Sciences, Komrat/ MOLDOVA. 182-189.