WEKA İLE VERİ MADENCİLİĞİ SÜRECİ VE ÖRNEK UYGULAMA

Pınar TAPKAN
Lale ÖZBAKIR
Adil BAYKASOĞLU

ÖZET

Veri madenciliği mevcut veriyi kullanışlı bilgiye çevirme ihtiyacından ortaya çıkmış, büyük boyuttaki veritabanlarından önceden bilinmeyen, gizli, anlamlı ve yararlı bilgilerin elde dilmesi süreci olup gelecek ile ilgili tahminde bulunmayı sağlar. Veri madenciliği sürecinin en önemli aşamaları ise verinin hazırlanması ve belirlenen amaca göre veri madenciliği algoritmalarının kullanımını içermektedir. Bu çalışmada veri önişleme ve sınıflandırma, kümeleme ve birliktelik kuralları algoritmaları incelenerek WEKA paket programında uygulamaları gerçekleştirilmiştir.

Anahtar Kelimeler: Veri madenciliği, Veri Önişleme, Sınıflandırma, Kümeleme, Birliktelik kuralları, WEKA.

ABSTRACT

Data mining arise from the requirement of transformation of the current knowledge to the practical and useful knowledge. It is the process of attaining unknown, hidden, meaningful and beneficial information from large sized databases which also enables to predict about feature. The most important stages of data mining process are data preparation and utilizing the data mining algorithms according to a predetermined purpose. In this research, preprocessing and data mining algorithms as classification, clustering and association rules are analyzed and the implementation of these methods by WEKA software is realized.

Key Words: Data mining, Preprocessing, Classification, Clustering, Association Rules, WEKA.

1. GİRİŞ

Veri toplama kaynaklarının ve bilişim teknolojisinin hızlı gelişimi neticesinde kurumlar üretmiş oldukları büyük boyutlardaki verilerden anlamlı ve yararlı bilgiler ortaya çıkarmakta zorluklar yaşamaktadırlar. Diğer taraftan geleneksek istatistiksel yöntemler, bu tür verileri çözümlemekte yetersiz kalmaktadır [1]. Temel olarak verileri işlemek ve çözümlemek için kullanılan veri madenciliği yöntemi, hemen her disiplin tarafından çeşitli amaçlar için kullanılan ve araştırmacılar tarafından takip edilen bir alan haline gelmiştir. Veri madenciliği, geçerli, uygulanabilir ve daha önce bilinmeyen bilgilerin büyük boyutlu veritabanlarından elde edilmesi ve karar aşamasında etkin olarak kullanılması olarak tanımlanabilir. Geniş veritabanlarının oluşturulmasına olanak veren bankacılık, pazarlama, sigortacılık, elektronik ticaret, sağlık gibi birçok alanda uygulama imkânı olan veri madenciliği uygulamalarına örnek olarak, müşterilerin satın alma alışkanlıklarının belirlenmesi, mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması, kredi kartı dolandırıcılıklarının, kredi kartı harcamalarına göre müşteri gruplarının ve riskli müşteri gruplarının belirlenmesi verilebilir.

Bu çalışmada temel olarak sınıflandırma, kümeleme ve birliktelik kuralları olarak adlandırılan veri madenciliği algoritmalarının detayları verilerek WEKA paket programı üzerinde uygulamalarına değinilecektir. Ancak ilgili algoritmaların uygulanabilmesi için eldeki verilerin bir önişlem aşamasından

geçirilmesi gerekebilmektedir. Bu kapsamda öncelikle verilerin yeniden yapılandırılması süreci değerlendirilecektir.

2. VERİNİN YENİDEN YAPILANDIRILMASI

Bazı uygulamalarda eldeki veri kümesi kayıp veriler, yanlış veriler, aşırı uç değerler, gereksiz bilgiler ya da değişken değerlerinin birbirinden ayrık olduğu durumları içerebilir. Ayrıca farklı veri madenciliği algoritmaları farklı veri tipleri gerektirebildiği için bu tür durumlarda verinin bir önişleme tabi tutulması gerekir.

2.1. Kayıp Veriler

Veri madenciliği algoritmasının uygulanacağı veri kümesinde bazı kayıtlar hiç girilmemiş olabilir. Kayıp verilere sahip bir veritabanına uygulanabilecek yöntemlerden ilki kayıp verinin bulunduğu kaydı veritabanından çıkarmaktır. Eğer kayıp verili kayıt sayısı, toplam kayıt sayısına göre oldukça az ise bu kayıtların veritabanından çıkarılması mümkündür. Diğer taraftan kayıp veri sayısı yüksekse veya bu kayıtlara ait diğer değerler önemliyse kayıp değerlerin yerine genel bir sabit kullanılabilir ya da kayıp verilerin yerine tüm verilerin ortalama değeri kullanılabilir [2]. Daha etkin olarak ise regresyon kullanılarak diğer değişkenlerin yardımı ile kayıp verilerin değerleri tahmin edilebilir. Ayrıca zaman serileri analizi, bayesyen sınıflandırma, karar ağaçları, maksimum beklenti gibi teknikler de kayıp verilerin tahmininde kullanılabilir.

2.2. Yanlış ya da Aşırı Uç Veriler

Kullanılan veritabanının hatalı ya da tutarsız değerler içermesine örnek olarak yaş bilgisinin 205 olması verilebilir. Böyle bir bilginin var olması mümkün değildir ve bu tür verilere gürültülü veri denir. Bu tür veriler için veri düzgünleştirme tekniği kullanılır. Tutarsız verilerin yaratacağı sorunları ortadan kaldırmanın en temel yolu bu uç değerlerin veritabanından çıkarılmasıdır. Diğer taraftan veri düzgünleştirme, basit olarak verilerin eşit boyuttaki alt kümelere ayrılması, her kümenin aritmetik ortalamasının alınması ve küme içindeki verilerin bu aritmetik ortalamayla değiştirilmesi esasına dayanır. Kullanılabilecek bir başka yöntem ise kenardaki verilerin birbirlerinden farkının küme elemanı sayısına bölünmesiyle elde edilen değerin o küme elemanına atanmasını içerir [2].

2.3. Gereksiz Veriler

Aynı veritabanı içerisinde hem yaş hem de doğum tarihi bilgisinin verilmesi gibi bilgisayar çalışma zamanını artıran ve elde edilecek sonuçların güvenilirliğini ve kalitesini etkileyebilecek veriler gereksiz veri olarak adlandırılır. Bu durumda mevcut değişkenlerin birleştirilmesi yani veri boyutunun indirgenmesi en akılcı yoldur. Bu amaçla en sık kullanılan yöntemler dalga dönüşümü ve temel bileşenler analizidir [2].

2.4. Değişken Değerlerinin Birbirinden Ayrık Olduğu Durumlar

Değişkenlerin ortalama ve varyansları birbirlerinden önemli ölçüde farklı olduğu takdirde büyük ortalama ve varyansa sahip değişkenlerin diğerleri üzerindeki baskısı daha fazla olur ve rollerini önemli ölçüde azaltır. Bu nedenle bir dönüşüm yöntemi uygulayarak söz konusu değişkenlerin normalleştirilmesi ve standartlaştırılması uygun bir yol olacaktır [1]. Bu amaçla en sık kullanılan yöntemler verinin alabileceği minimum ve maksimum değeri kullanan min-maks normalizasyonu ve ortalama ve standart sapma değerlerini kullanan sıfır-ortalama normalizasyonudur.

2.5. Kullanılacak Algoritmaya Bağlı Yapılandırmalar

Veri madenciliğinde kullanılan bir takım teknik ve algoritmalar sadece bazı türdeki verilerle çalışabilirler. Bazı algoritmalar sadece sayısal değerlerle, bazıları sadece kategorik değerlerle bazıları ise 0-1 değerlerle işlem yaparlar. Bu durumda mevcut veri kullanılacak algoritmaya uygun hale getirilmelidir.

3. SINIFLANDIRMA

Ele alınan veri madenciliği algoritmalarından ilki olan sınıflandırma, temel olarak veritabanındaki gizli örüntülerin ortaya çıkarılması yani verinin ortak özelliklerine göre ayrıştırılması için kullanılır. Sınıflandırma bir öğrenme algoritmasına dayanır, öncelikle veritabanının bir kısmı örnek veri kümesi olarak belirlenerek eğitim amacıyla kullanılır ve sınıflandırma kuralları oluşturulur, daha sonra bu kurallar yardımıyla yeni bir durum ortaya çıktığında nasıl karar verileceği belirlenir. Böylece hangi sınıfa ait olduğu bilinmeyen bir kayıt için bir sınıf belirlenebilir.

3.1. Karar Ağaçları ile Sınıflandırma

Karar ağaçları ile sınıflandırma temel olarak bir karar ağacının oluşturulması, veritabanındaki her kaydın bu ağaca uygulanması ve çıkan sonuca göre kaydın sınıflandırılmasına dayanır. Diğer yöntemlerle karşılaştırıldığında karar ağaçlarının yapılandırılması ve anlaşılması daha kolaydır. Karar ağacı oluşturulurken kullanılan algoritmaya göre ağacın şekli değişmekte ve değişik ağaç yapıları farklı sınıflandırma sonuçları verebilmektedir. Karar ağaçlarına bağlı olarak geliştirilen birçok algoritma olup bu algoritmalar kök, düğüm ve dallanma kriteri seçimlerinde izledikleri yol açısından farklılık göstermektedirler.

Karar ağaçlarının oluşturulması sırasında dallanmaya hangi nitelikten başlanacağı oldukça önemlidir çünkü olası tüm ağaç yapılarını ortaya çıkararak içlerinden en uygun olanı ile başlamak mümkün değildir. Bu sebeple karar ağacı algoritmalarının çoğu daha başlangıçta birtakım değerleri hesaplayarak ona göre ağaç oluşturma yoluna gitmektedir. Bu hesaplamalardan biri de entropiye dayalı olup, entropi belirsizliğin ölçüsü olarak tanımlanmaktadır. Entropi, bir veri kümesi içindeki belirsizlik ve rastgeleliği ölçmek için kullanılır ve 0 ile 1 arasında değer alır. Bütün olasılıklar eşit olduğunda entropi maksimum değerini alacaktır [1]. Entropiye dayalı karar ağaçları ile sınıflandırma algoritmalarının en önemlileri aşağıdaki gibidir.

ID3 [3]: ID3, makine öğrenme ve bilişim teorisine bağlı olarak verilen örnekler içinde en ayırıcı değişkeni bulan bir algoritmadır. Temel olarak kategorik nitelikleri sınıflandırır ve veritabanı dallandırılmadan önce ve sonra doğru sınıflandırma yapmak için gelen bilgiler arasındaki farkı kullanarak, öncelikli düğüme ve dallanmalara karar verir.

C4.5 [4]: ID3 algoritmasından farklı olarak sayısal değerlere sahip niteliklerin karar ağaçlarının oluşturulmasını sağlar. Diğer taraftan karar ağacı oluştururken kayıp verileri almaması sebebiyle daha anlamlı kurallar sunan ağaçlar üretebilir. Kayıp veriler ise diğer veri ve değişkenler kullanılarak tahmin edilir.

CART [5]: CART algoritması, her karar düğümünden sonra ağacın iki dala ayrılması ilkesine dayanır. Bu teknikte dallanma kriteri belirlenirken kayıp veriler önemsenmez.

3.2. Yapay Sinir Ağları ile Sınıflandırma

Yapay sinir ağları ile sınıflandırmanın işleyiş yapısı, çıktı katmanına ulaşabilmek için ağırlıkların hesaplanmasına dayanır. Eğitim veri kümesi üzerinde hesaplanan ağırlıklar, test veri kümesi üzerinde kullanılarak öğrenmenin ne kadar gerçekleştiği belirlenir. Elde edilen ağırlıkların etkinliği

doğrulanamazsa ağırlıklar üzerinde düzeltme ve yeniden hesaplama işlemleri gerçekleştirilir. Öğrenme süreci tamamlandığında ise ağırlıklar yardımıyla yeni bir verinin hangi sınıfa ait olduğu belirlenebilir. Yapay sinir ağlarında öğrenme süreci uzun sürse de oldukça duyarlı sınıflandırmalar yapabilmektedir.

3.3. İstatistiğe Dayalı Algoritmalar

Regresyon analizi, lojistik regresyon, zaman serileri analizi ve Bayesyen yaklaşımı gibi istatistiksel yöntemler veri madenciliğinde sınıflandırma algoritması olarak kullanılmaktadır. Bunlardan en sık kullanılanları sınıflanmış verileri kullanarak yeni bir verinin mevcut sınıflara girme olasılığını hesaplayan Bayesyen sınıflandırma algoritması ve regresyon analizidir.

3.4. Mesafeye Dayalı Algoritmalar ile Sınıflandırma

Mevcut verilerin birbirlerine olan uzaklıkları ve benzerliklerine dayanan bu tür algoritmaların en yaygın kullanılanı k-en yakın komşu algoritmasıdır. Bu algoritmanın amacı, sınıfları belli bir örnek kümedeki gözlem değerlerini kullanarak yeni gözlemlerin hangi sınıfa ait olduğunu belirlemektir. Örnek kümedeki gözlemlerin her birinin, sonradan belirlenen bir gözlem değerine olan uzaklıklarının hesaplanması ve en küçük uzaklığa sahip k adet gözlemin seçilmesi esasına dayanır. Bilimsel yazında k-en yakın komşu algoritmasını temel alan birçok algoritma geliştirilmiştir.

4. KÜMELEME

Kümeleme, verilerin birbirlerine olan benzerliklerine göre gruplandırılmasına dayanır. Sınıflandırmada olduğu gibi bir gruplandırma söz konusu olsa da sınıflandırmadan farklı olarak sınıflar önceden belli değildir. Kümeleme algoritmaları, küme oluşturma stratejisine ve kullanılan veri türüne göre farklılık gösterirler. Kümeleme yöntemlerinin çoğu veriler arasındaki uzaklıkları yani veriler arasındaki benzerlik ya da farklılıkları kullanırlar.

4.1. Hiyerarşik Algoritmalar

Veritabanındaki her noktanın birleşiminden oluşan bir kümenin aşamalı olarak alt kümelere ayrılması (bölünür kümeleme algoritmaları) ya da veritabanındaki her noktanın ayrı bir küme olarak ele alınarak bu kümelerin birleştirilmesiyle ayrı kümelere ulaşılması (toplaşım kümeleme algoritmaları) esasına dayanır. Kullanımı kolay ve hemen hemen tüm veri tiplerine uygulanabilen esnek bir yapıya sahiptir.

SLINK (En yakın komşu algoritması) [6]: Her bir verinin ayrı bir küme olarak ele alındığı ve aşamalı olarak bu kümelerin birleştirildiği bir yapıya sahiptir. Bu algoritmada iki kümenin birbirine olan uzaklığı, o kümelerdeki birbirine en yakın verilerin birbirine olan uzaklığı olarak kabul edilir. Eğer eldeki uzaklık verisi belli bir eşik değerini geçiyorsa kümeler birleştirilir.

CURE (Temsilciler kullanarak kümeleme) [7]: Veritabanı içinde diğer verilerden uzakta bulunan ve sayıları az olup aslında hiç bir kümeye ait olmaması gereken uç verilerin kümeleme kalitesini etkilememesi amacıyla geliştirilmiş bir algoritmadır. En yakın komşu algoritmasındaki toplaşım ve yakınlık prensibine dayanır.

En uzak komşu algoritması: En yakın komşu algoritmasından farklı olarak iki kümenin birbirine olan uzaklığı, kümelerdeki birbirine en uzak verilerin arasındaki uzaklıkla belirlenir.

CHAMELEON [8]: İki kümenin birbirine olan uzaklığının yanı sıra birbirine olan benzerlikleri bilgisini de kullanır. İki kümenin birleştirilmesi esnasında, kümelerin birbirine olan benzerliği ve yakınlığı ile bu kümelerin kendi iç benzerlikleri ve yakınlıkları karşılaştırılır. Böylece daha kaliteli ve homojen kümeler elde edilir. En yakın komşu algoritmasındaki toplaşım ve yakınlık prensibine dayanır.

BIRCH (Hiyerarşi kullanarak dengelenmiş iteratif azaltma ve kümeleme) [9]: Temel olarak gürültülü verilerin kontrol edilmesi amacıyla büyük boyutlu veritabanlarının kümelenmesi için geliştirilmiştir. En uzak komşu algoritmasında olduğu gibi bölünür bir yapıya sahip olan algoritma, sadece sayısal verilere uygulanabilmektedir.

4.2. Bölümlemeli Algoritmalar

Kümeler arasındaki minimum ya da maksimum uzaklığın, kümelerin iç benzerlik kriterlerinin ve küme sayısının kullanıcı tarafından belirlendiği algoritmalardır. Hiyerarşik algoritmalardan daha hızlı çalışan bölümlemeli algoritmalar, bu özelliklerinden dolayı büyük veritabanlarının kümelenmesi için daha uygundur.

k-ortalama algoritması [10]: Verilerin kümelerin ortalamalarına göre önceden belirlenmiş k adet kümeye ayrılması esasına dayanan bir algoritmadır. Toplam ortalama hatanın minimize edilmesini amaçlayan algoritma sadece sayısal verilerde kullanılabilir. Bu alandaki algoritmaların çoğu k-ortalama algoritmasının geliştirilmesiyle ortaya çıkmıştır.

k-medoid algoritması [11]: Sayısı önceden belirlenmiş k adet kümenin her biri için k adet medoid belirlenmesi ile başlayan algoritma veritabanındaki diğer verilerin kendilerine en çok benzeyen medoidlerin etrafına toplanması esasına dayanır. Medoid ise kümenin merkezine yakın uzaklıkta bulunan noktayı temsil etmektedir.

CLARA algoritması (Geniş uygulamaların kümelenmesi) [11]: k-medoid algoritmasından farklı olarak tüm veritabanını tarayarak medoid noktaları belirlemek yerine veritabanından rastgele oluşturulan bir örnek küme üzerinde benzer şekilde çalışır. İki algoritma karşılaştırıldığında CLARA algoritmasının büyük boyutlu veritabanları için daha güvenli olduğu ve daha kısa süre içinde kümeleme yapabildiği belirtilmiştir.

CLARANS algoritması (Rastgele aramaya dayalı geniş uygulamaları kümeleme) [12]: k-medoid ve CLARA algoritmalarının geliştirilmiş bir halini barındıran CLARANS algoritması şebeke diyagramından yararlanan bir yapıya sahiptir. CLARA algoritmasına benzer olarak bütün veritabanı taranmazken yapılan örnekleme dinamik bir yapıya sahiptir.

4.3. Yoğunluğa Dayalı Algoritmalar

Dağılmış verilere sahip veritabanlarının sadece uzaklığı temel alan bölümlemeli algoritmalar ile kümelenmesi oldukça güçtür. Çünkü hiç bir kümeye dâhil olmayan uç noktalar içeren bu dağılmış veritabanlarının bölümlemeli algoritmalar ile kümelenmesi neticesinde doğru kümeler ortaya çıkmayacaktır. Bu durumda birlikte bir yoğunluk oluşturan verilerin aynı kümeye alınmasına dayanan yoğunluğa dayalı algoritmalar kullanılmalıdır. Bu tür algoritmalara örnek olarak DBSCAN, OPTICS ve DENCLUE algoritmaları verilebilir.

4.4. Grid Temelli Algoritmalar

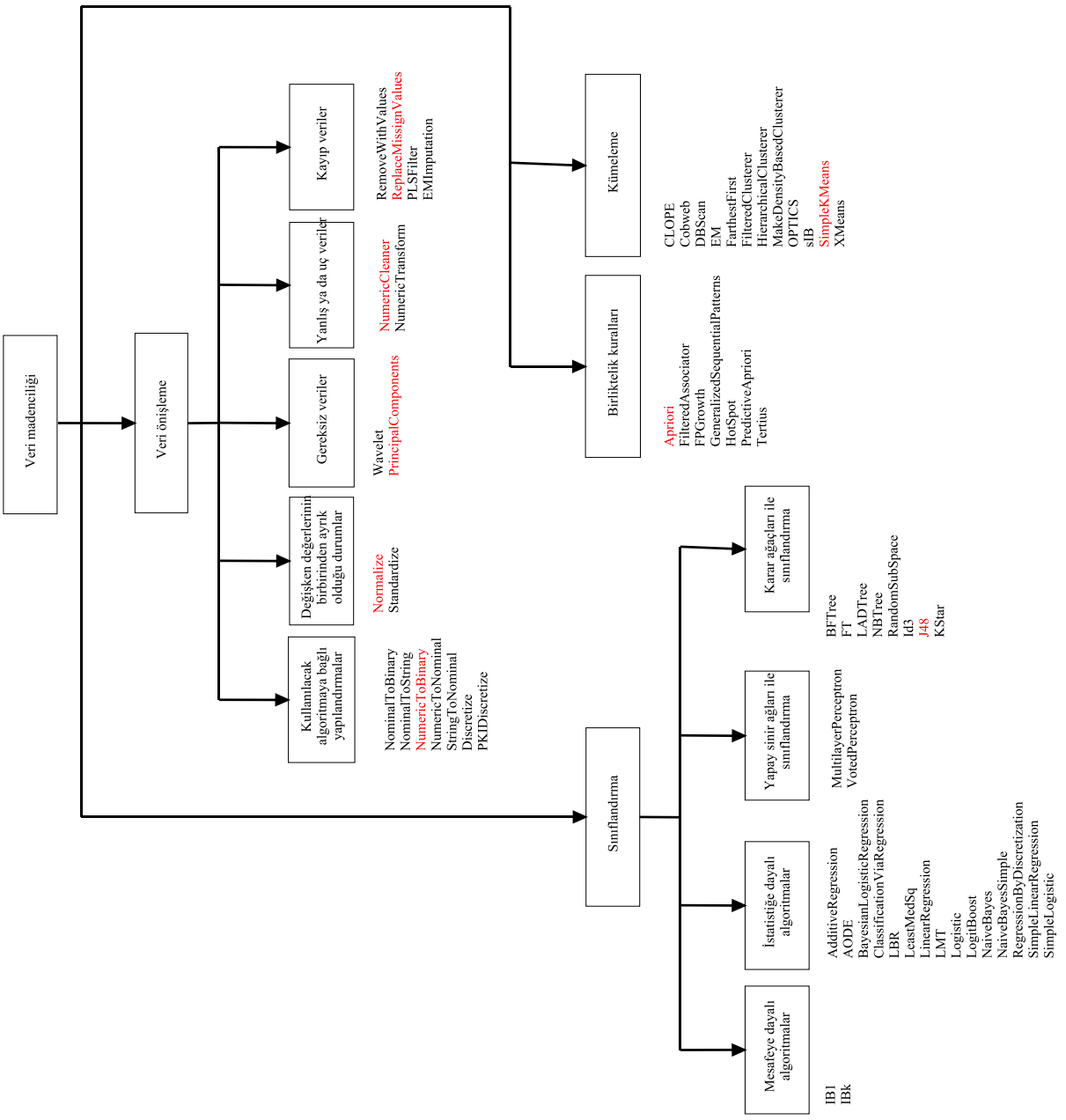
Büyük boyuttaki veritabanlarının kümelenmesinde numaralandırılmış çizgilerden oluşan hücresel yapıları kullanan algoritmalardır. Bu algoritmalara örnek olarak ise bölgenin dikdörtgen hücrelere bölünerek hiyerarşik bir yapının kullanıldığı STING algoritması, değişik şekillerde kümeler sunabilen ve hassas kümeleme kabiliyeti olan dalga kümeleme algoritması ve hem yoğunluğa hem de grid yapısına sahip CLIQUE algoritması verilebilir.

5. BİRLİKTELİK KURALLARI

İncelemekte olduğumuz son veri madenciliği tekniği olan birliktelik kuralları, olayların birlikte gerçekleşme durumlarını inceleyen bir yöntemdir. Genel olarak müşterilerin satın alma eğilimlerini belirlemek amacıyla kullanılır. Teorik olarak kullanıcı tarafından belirlenen en küçük destek ve en küçük güven seviyelerinin üstünde destek ve güven seviyesine sahip kuralların belirlenmesidir. Birliktelik kurallarının oluşturulması amacıyla yaygın olarak kullanılan algoritmalara örnek olarak geniş nesne kümeleri üretmek için geliştirilmiş ve veritabanının birçok kez taranmasını gerektiren AIS, SETM ve APRIORI algoritmaları; veritabanının tamamının taranmasına gerek duymayan AprioriTid algoritması; APRIORI ve AprioriTid algoritmalarının birleşimini içeren Apriori-Hybrid algoritması, geniş nesne kümelerini belirlemek için veritabanından alınan küçük örneklerin kullanımının yeterli olduğu fikrine dayanan OCD algoritması verilebilir.

6. WEKA

WEKA (Waikato Environment for Knowledge Analyses), Waikato Üniversitesi tarafından geliştirilerek 1996'da ilk resmi sürümü yayınlanmış olan bir makine öğrenme ve veri madenciliği yazılımıdır. Akademik araştırmalar, eğitim ve endüstriyel uygulama alanlarında kullanım yeri olan WEKA, veri analizi ve tahminleyici modelleme için geliştirilmiş algoritma ve araçların görsel bir birleşimini içerir. Geliştirilen yazılımın temel avantajları geniş veri önişleme ve modelleme tekniklerine sahip olması, grafiksel kullanıcı arayüzü sayesinde kullanımının kolay olması ve Java programlama dili ile uygulandığından herhangi bir platformda kullanılabilmesi yani taşınabilir olmasıdır. Önceki bölümlerde bahsedilen veri önişleme ve veri madenciliği algoritmaları çerçevesinde WEKA'nın hiyerarşik yapısı Şekil 1'de sunulmuştur.



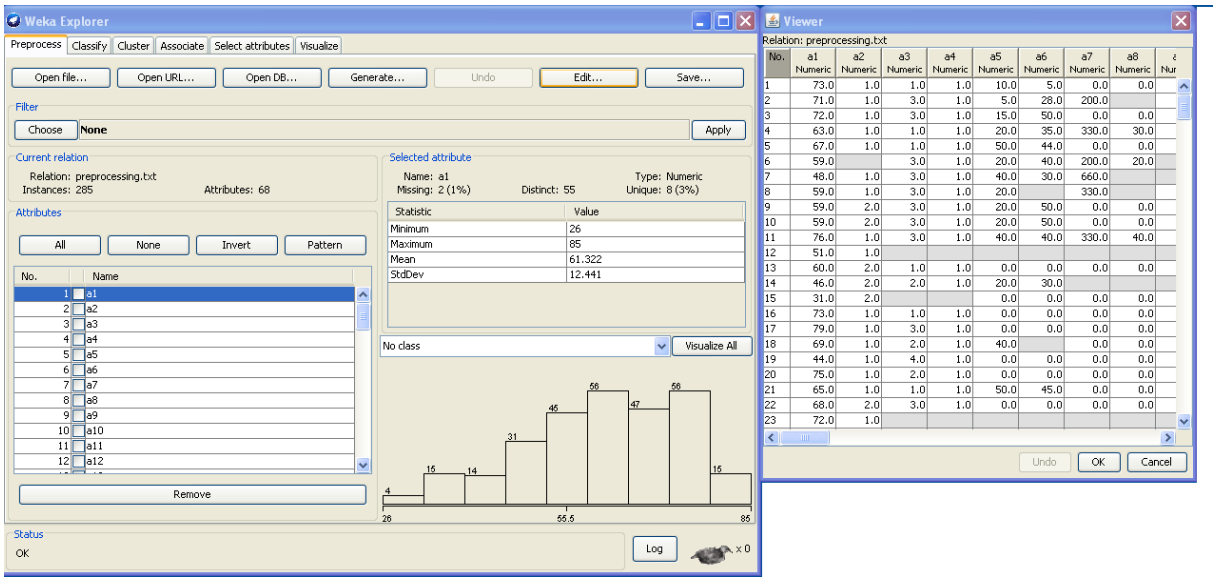
Şekil 1. WEKA Hiyerarşik Yapısı

Şekil 1'de veri önişleme ve her bir veri madenciliği algoritmasının WEKA'daki kullanımına ilişkin örnek modüller verilmiştir. Çalışmanın bu bölümünde her yöntem için seçilen bir modül uygulamalı olarak anlatılacaktır. Bu uygulamalar esnasında kullanılacak veri kümesi [13] 285 adet örnek mide kanseri verisi içermekte olup 9 sınıf ve 7 adeti nümerik kalanları ise kategorik olmak üzere 68 niteliğe sahiptir. Veritabanı içerisinde 970 adet kayıp veri bulunmakta olup bütün veritabanı içerisinde %5'lik bir belirsizlik söz konusudur.

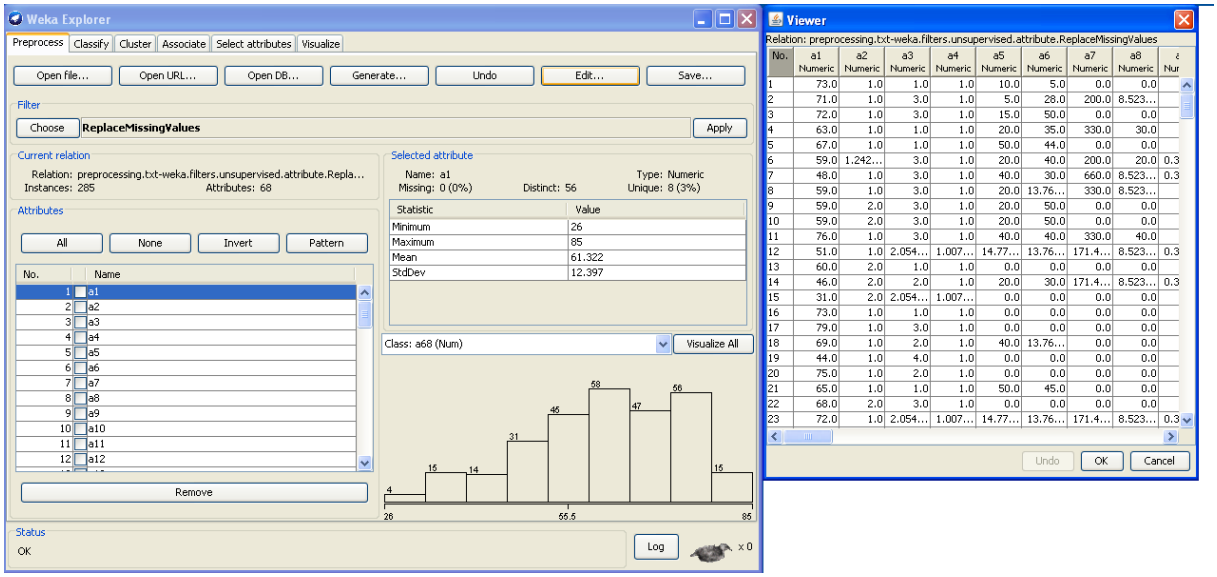
6.1. Veri Önişleme

6.1.1. Kayıp Veriler

Kayıp verilerin yaratacağı sorunları ortadan kaldırmak için kullanılan yöntemlere örnek olarak ReplaceMissingValues modülü kullanılmıştır (Şekil 2-3). Bu yöntemle veritabanındaki kayıp değerler, ait oldukları niteliğin diğer değerlerinin ortalaması ya da moduyla değiştirilmektedir.



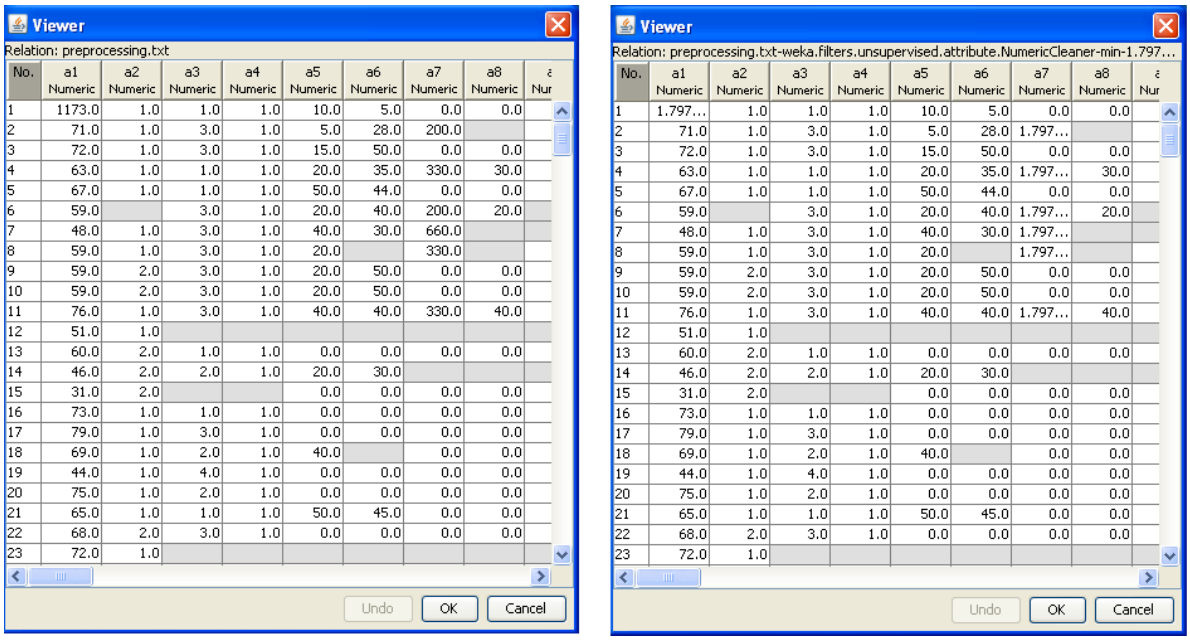
Şekil 2. Mevcut Veritabanındaki Kayıp Veriler



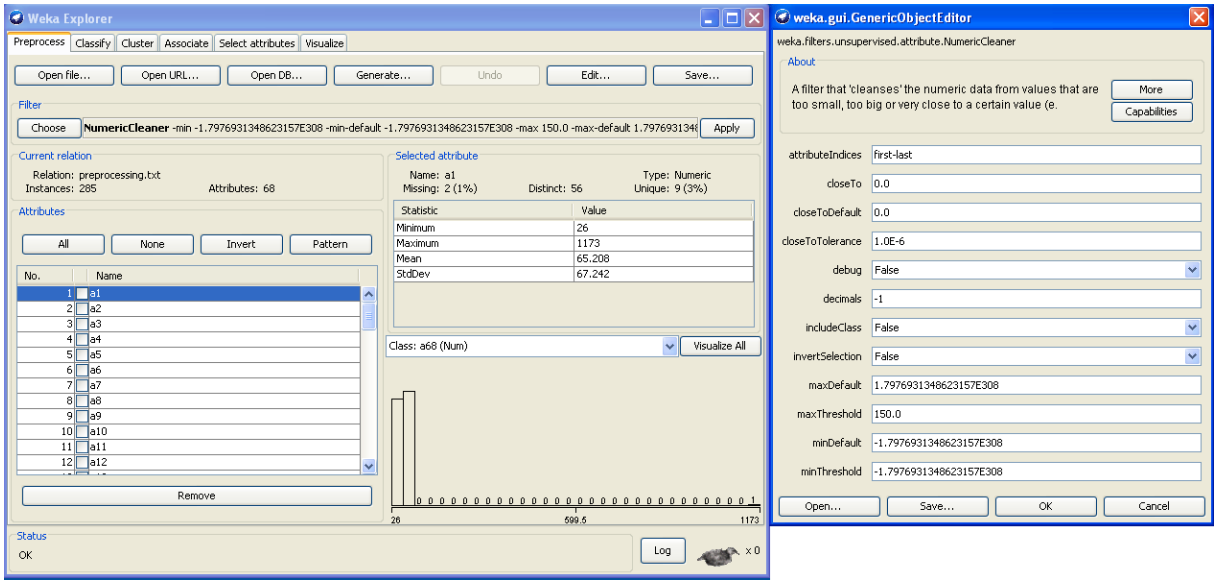
Şekil 3. ReplaceMissingValues Modülünün Kullanımı

6.1.2. Yanlış ya da Aşırı Uç Veriler

Bu tür veriler için ise NumericCleaner modülü ele alınmıştır (Şekil 4-5). Bu yöntem çok büyük, çok küçük ya da belli bir değere çok yakın değerlerin veritabanından silinerek bu değerlerin yerine önceden belirlenmiş başka bir değerin atanmasını içerir.



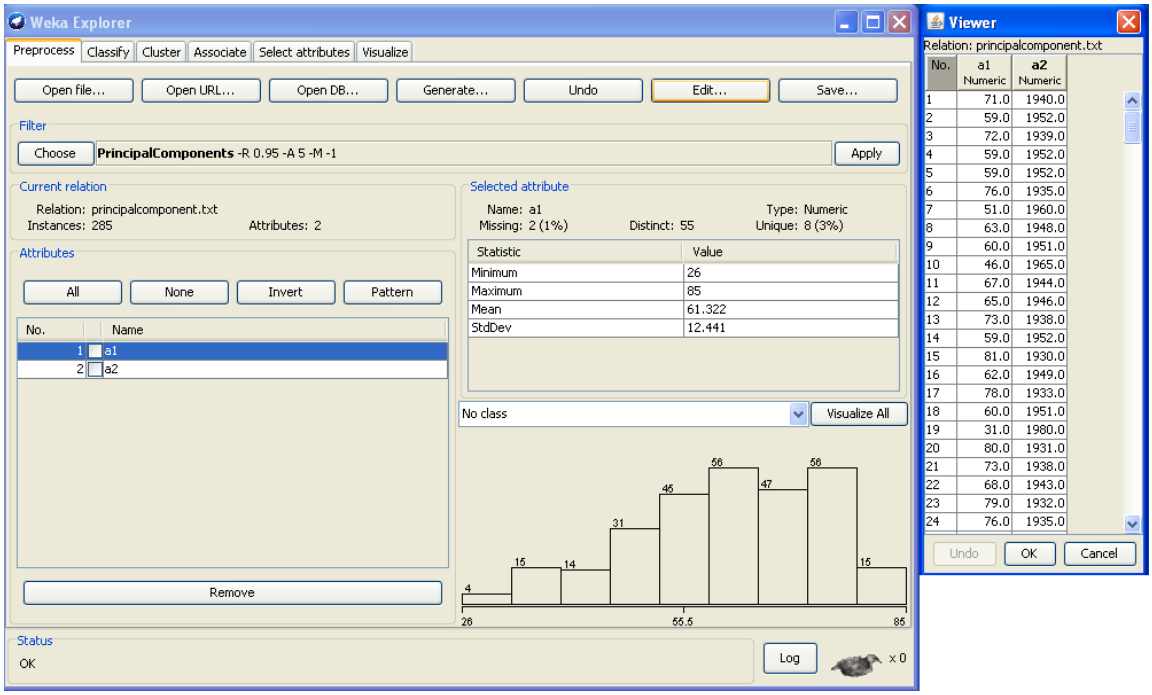
Şekil 4. Aşırı uç verilerin NumericCleaner Modülü Kullanılmadan Önce ve Sonraki Durumu



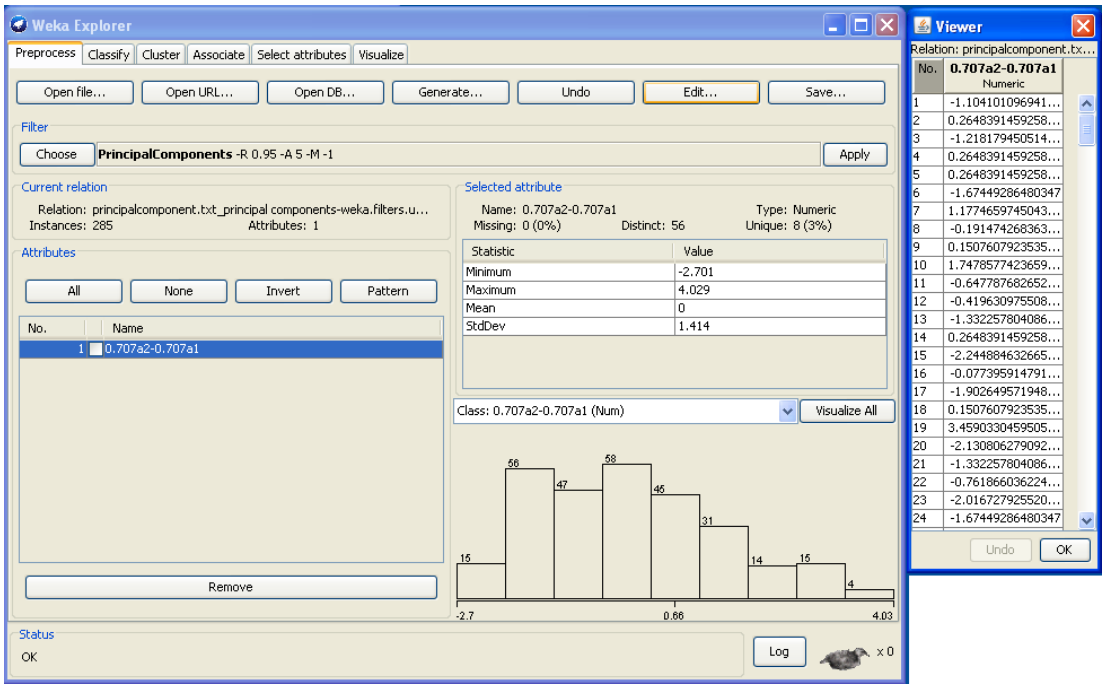
Şekil 5. NumericCleaner Modülünün Kullanımı

6.1.3. Gereksiz Veriler

Aynı veritabanı içinde hem yaş hem de doğum tarihi bilgisinin verilmesi durumunda oluşan gereksiz verilerin bilgisayar çalışma zamanını ve sonuçların kalitesini etkilememesi amacıyla PrincipalComponents modülü kullanılarak veri boyutu azaltılmıştır (Şekil 6-7).



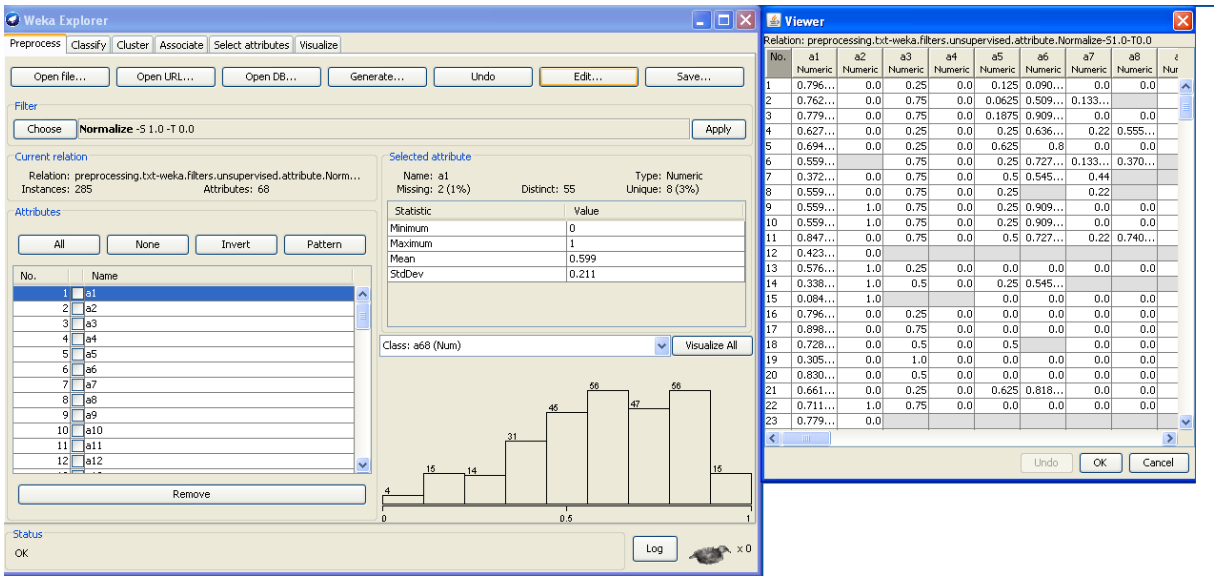
Şekil 6. PrincipalComponents Modülü Kullanılmadan Önceki Durum



Şekil 7. PrincipalComponents Modülü ile Boyut İndirgeme

6.1.4. Değişken Değerlerinin Birbirinden Ayrık Olduğu Durumlar

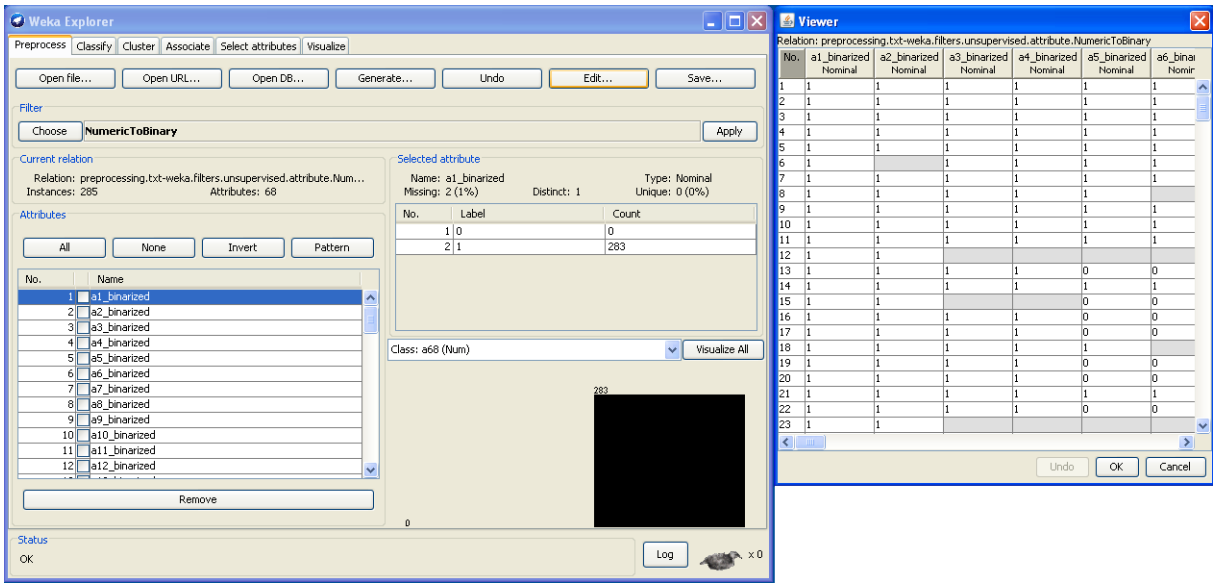
Değişken değerlerinin birbirinden ayrık olduğu durumlar için WEKA'nın Normalize modülünden faydalanılmıştır (Şekil 8). Bu yöntemle nümerik değerler birbirine yaklaştırılarak normalleştirilmektedir.



Şekil 8. Normalize Modülünün Kullanımı

6.1.5. Kullanılacak Algoritmaya Bağlı Yapılandırmalar

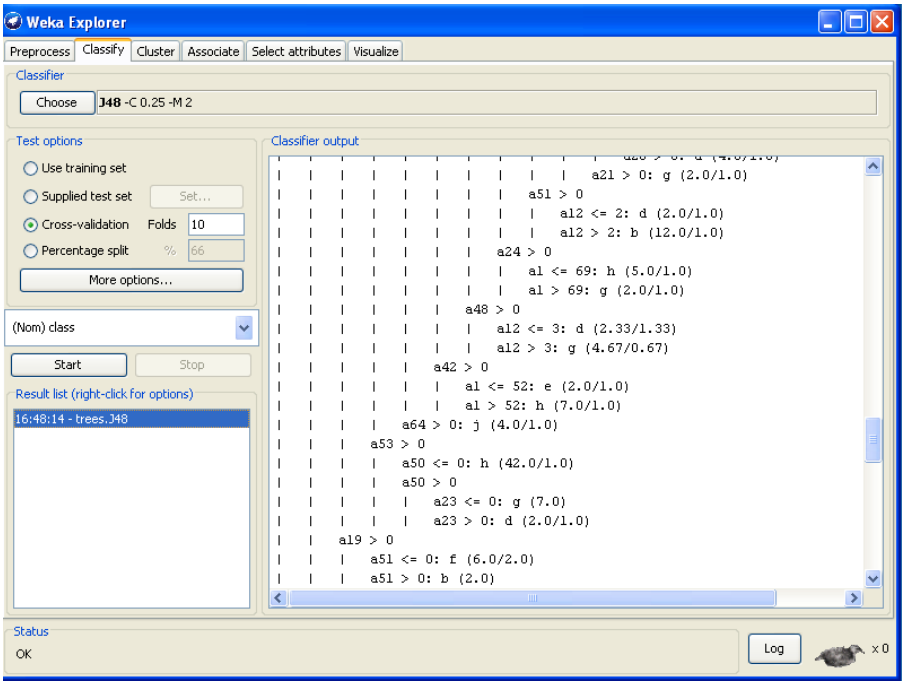
Veri önişlemeye gereksinim duyan diğer bir durum olan algoritmanın sebep olduğu kısıtlar için ise veritabanındaki nümerik değerlerin 0-1 değerlere dönüştürülmesi esasına dayanan NumericToBinary modülü kullanılmıştır (Şekil 9).



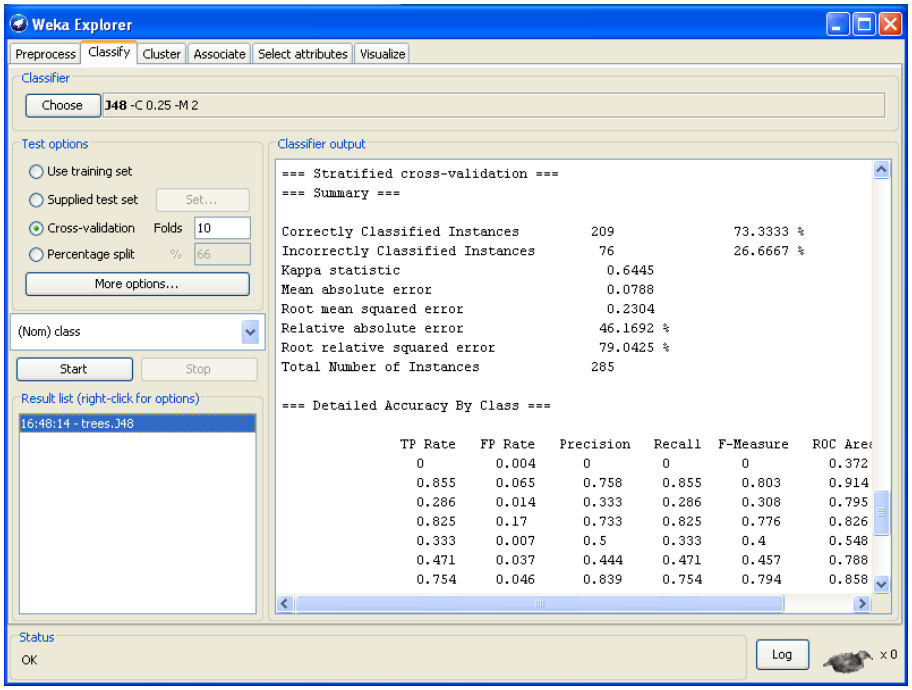
Şekil 9. NumericToBinary Modülünün Kullanımı

6.2. Sınıflandırma

Sınıflandırma algoritmaları içerisinden Bölüm 3.1'de detayları verilen C4.5 karar ağacına dayanan J48 modülü, uygulaması gerçekleştirilmek üzere seçilmiştir (Şekil 10-11).

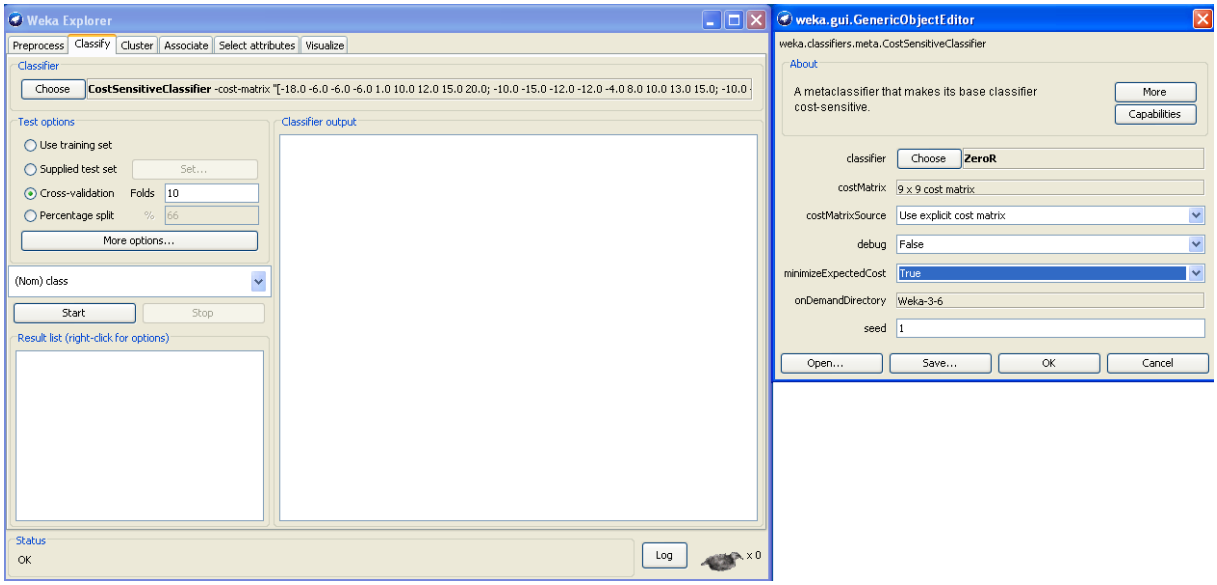


Şekil 10. J48 Modülünün Kullanımı ile Elde Edilen Kurallar

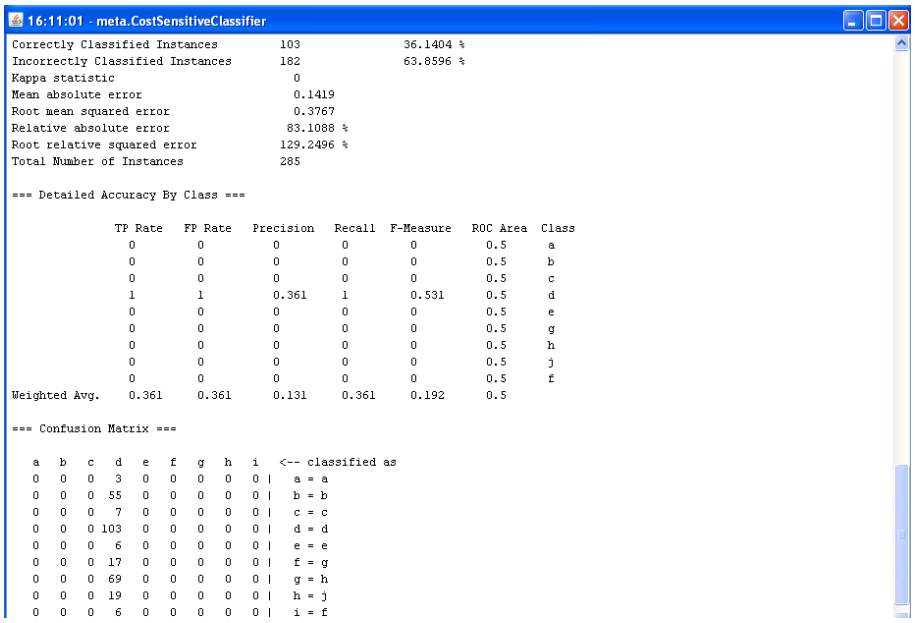


Şekil 11. J48 Modülü ile Elde Edilen Sınıflandırma Doğrulukları

Verilerin sınıflandırılması esnasında göz önünde bulundurulması gereken noktalardan biri de maliyetlerdir. Yanlış sınıflandırmanın bir maliyete tabi olduğu maliyete göre sınıflandırma yöntemlerinde belli bir hatalı sınıflandırmanın göreli önemi diğer hatalı sınıflandırmalardan daha fazla olabilmektedir. Bu bağlamda WEKA'da maliyete göre sınıflandırma yapan CostSensitiveClassifier modülü kullanılacaktır. Bu yöntemin amacı beklenen yanlış sınıflandırma maliyetini minimize edecek en iyi sınıflandırmayı tahmin etmektir (Şekil 12-13).



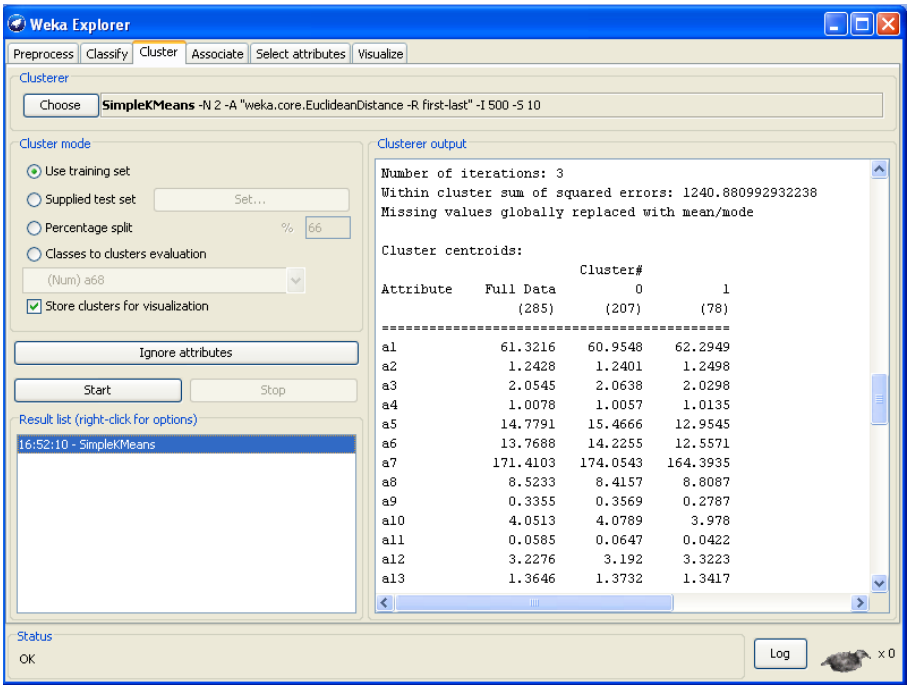
Şekil 12. CostSensitiveClassifier Modülünün Kullanımı



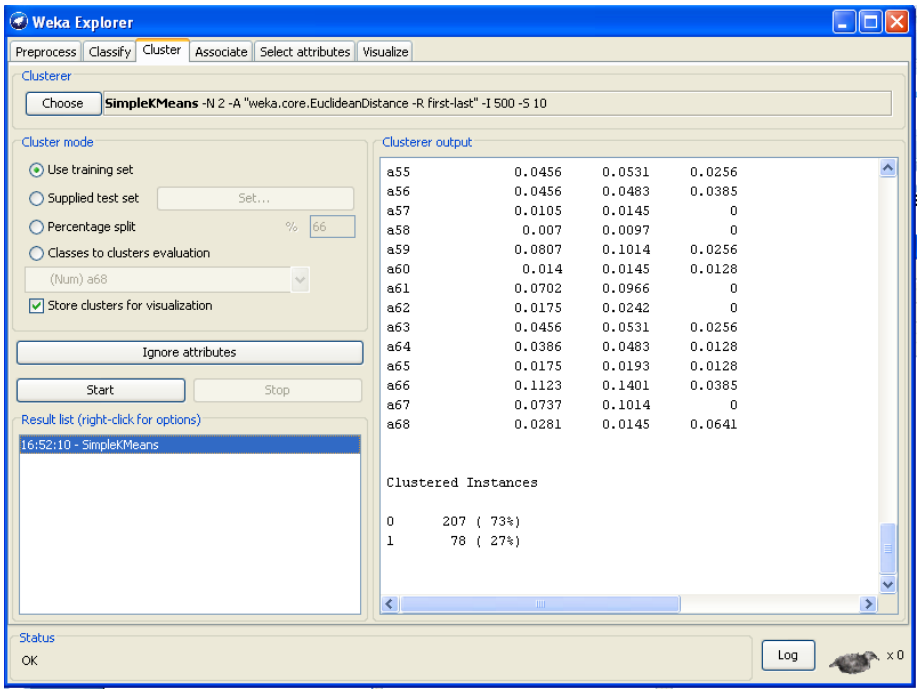
Şekil 13. CostSensitiveClassifier Modülünün Kullanımı ile Elde Edilen Sınıflandırma

6.3. Kümeleme

Kümeleme algoritmaları içinden ise bölümlemeli algoritmalar sınıfına giren ve birçok algoritmanın temel felsefesini oluşturan k-ortalama algoritmasını kullanan SimpleKMeans modülü ele alınmıştır (Şekil 14-15).



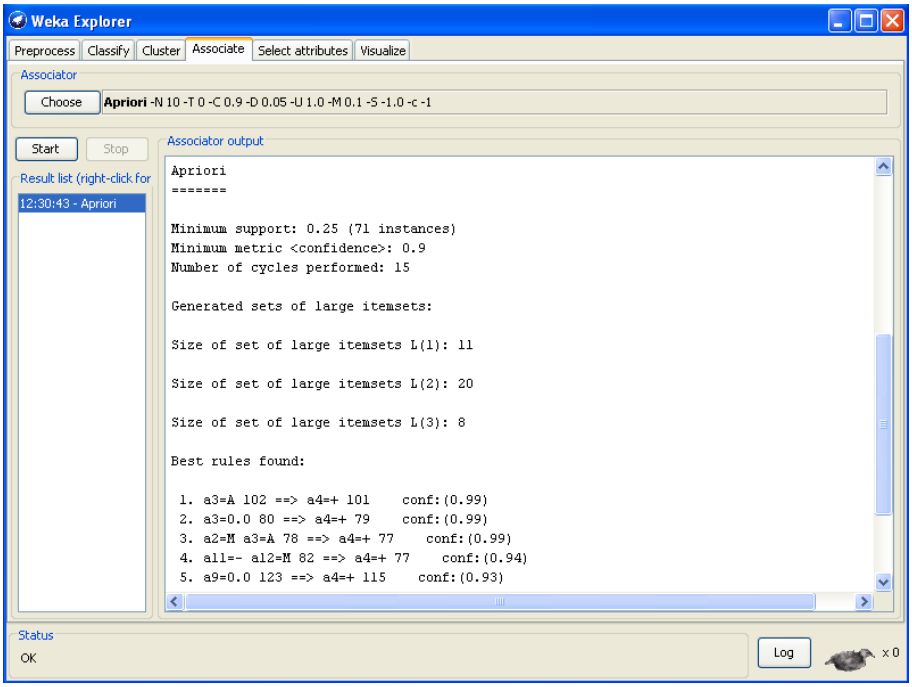
Şekil 14. SimpleKMeans Modülü ile Edinilen Kümeleme Bilgileri



Şekil 15. SimpleKmeans Modülünün Kullanımı Neticesindeki Kümeleme Yüzdeleri

6.4. Birliktelik Kuralları

Belli bir güven aralığı içinde istenen sayıda kuralı bulana kadar en küçük destek seviyesini aşamalı olarak azaltan Apriori algoritması WEKA içerisinde Apriori modülü ile ifade edilmekte olup uygulaması Şekil 16'daki gibi gerçekleştirilmiştir.



Şekil 16. Apriori Modülü ile Elde Edilen Birliktelik Kuralları

SONUÇ

Bu çalışmada veri madenciliğinin temel aşamaları olan veri önişleme ve veri madenciliği algoritmalarının kullanımı ele alınmıştır. İlgili süreçlerle ilgili temel bilgiler verilmiş, kullanılabilecek yöntemler tanıtılmış ve örnek bir veritabanı üzerinde seçilen yöntemlerin uygulaması WEKA yazılımı kullanılarak gerçekleştirilmiştir.

KAYNAKLAR

[1] ÖZKAN, Y., "Veri Madenciliği Yöntemleri", Papatya Yayıncılık, İstanbul, Türkiye, 2008.
[2] SİLAHTAROĞLU, G., "Kavram ve Algoritmalarıyla Temel Veri Madenciliği", Papatya Yayıncılık, İstanbul, Türkiye, 2008.
[3] QUINLAN, J.R., "Induction of Decision Trees ", Journal of Machine Learning, vol. 1, 81-106, 1986.
[4] QUINLAN, J.R., "C4.5: Program for Machine Learning", Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[5] BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A., STONE, C.J., "Classification and Regression Trees", Wadsworth, Belmont, 1984.
[6] SIBSON, R., "SLINK: An Optimally Efficient Algorithm for the Single Link Cluster Method", The Computer Journal, vol. 16(1), 30-34, 1973.
[7] GUHA, S., RASTOGI, R., SHIM, K., "CURE: An Efficient Clustering Algorithm for Large Databases", in: Proceedings of the 1998 ACM SIGMOD Intenational Conference on Management Data, vol. 27(2), 73-84, 1998.
[8] KARYPIS, G., HAN, E., KUMAR, V., "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling", IEEE Computer, vol. 32(8), 68-75, 1999.
[9] ZHANG, T., RAMAKRISHNAN, R., LIVNY, M., "BIRCH: An Efficient Data Clustering Method for Very Large Databases", in: Proceedings of the 1996 ACM SIGMOD International Conference on Management Data, vol. 25(2), 1996.

[10] MACQUEEN, J., "Some Methods for Classification and Analysis of Multivariate Observations", in: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 281-297, 1967.
[11] KAUFFMAN, L., ROUSSEEUW, P.J., "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley and Sons, 1990.
[12] RAYMOND, T.NG, HAN, J., "Efficient and Effective Clustering Methods for Spatial Data Mining", in: Proceedings of the 20th VLDB Conference, 144-155, 1994.
[13] GÜVENİR, H.A., EMEKSİZ, N., İKİZLER, N., ÖRMECİ, N., "Diagnosis of Gastric Carcinoma by Classification on Feature Projections", Artificial Intelligence in Medicine, vol. 31, 231-240, 2004.

ÖZGEÇMİŞ

Pınar TAPKAN

1979 yılı Kayseri doğumludur. 2000 yılında Erciyes Üniversitesi Endüstri Mühendisliği Bölümünü bitirmiştir. Bilkent Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalından 2002 yılında Yüksek Mühendis ve Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalından 2010 yılında Doktor ünvanını almıştır. 2001-2010 yılları arasında Araştırma Görevlisi olarak görev yapmıştır. 2010 yılından beri Erciyes Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümü'nde Yrd. Doç. Dr. olarak görev yapmaktadır. Yöneylem araştırma metodolojisi, optimizasyon kuramı, simülasyon teknikleri ve veri madenciliği ilgi alanlarıdır.

Lale ÖZBAKIR

1971 yılı Kayseri doğumlu Yrd. Doç. Dr. Lale Özbakır, Lisans öğrenimini 1992 yılında Bilkent Üniversitesi Bilgisayar ve Enformatik Mühendisliği Bölümünde tamamlamıştır. Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Yönetim ve Organizasyon Anabilim Dalında 1997 yılında yüksek lisans derecesini, Üretim Yönetimi ve Pazarlama Anabilim Dalında 2004 yılında doktora derecesini almıştır. Erciyes Üniversitesi Endüstri Mühendisliği Bölümüne 1997 yılında araştırma görevlisi, 2004 yılında Yardımcı Doçent olarak atanmış olup halen aynı bölümde Öğretim Üyesi olarak görev yapmaktadır. Yazarın uluslararası bilimsel dergilerde 30'un üzerinde, ulusal ve uluslararası kongrelerde 50'nin üzerinde bilimsel yayını bulunmaktadır. Yrd. Doç. Dr. Lale Özbakır çok sayıda ulusal ve uluslararası dergide hakemlik görevi yapmakta olup, çalışma alanları içerisinde veri madenciliği, yapay zeka ve meta-sezgisel yaklaşımlar, evrimsel algoritmalar, yöneylem araştırması yer almaktadır.

Adil BAYKASOĞLU

Prof. Dr. Adil Baykasoğlu Isparta Teknik Lisesi Makina bölümünden mezun olduktan sonra Lisans ve Yüksek Lisans derecelerini Makina Mühendisliği alanında 1993 ve 1995 yıllarında Gaziantep'te, doktora derecesini ise YÖK bursu ile gittiği Nottingham Üniversitesinden 1999 yılında Endüstri Mühendisliği alanında almıştır. 1993-2010 yılları arasında Gaziantep Üniversitesi Endüstri Mühendisliği Bölümünde çalışan Prof. Baykasoğlu halen Dokuz Eylül Üniversitesi Endüstri Mühendisliği bölümünde çalışmaktadır. Prof. Baykasoğlu ulusal ve uluslar arası bilimsel dergi ve kongrelerde 300 civarında bilimsel makale yayımladı. Yazarın ayrıca üç adet yayımlanmış kitabı, düzenleyip editörlüğünü yaptığı çeşitli ulusal ve uluslar arası kongre kitapları bulunmaktadır. Yazarın çalışma alanları genelde yöneylem araştırması, bilişimsel yapay zekâ, zeki etmenler, lojistik ve üretim sistemleri yönetimi/tasarımı, bilgisayar destekli üretim, kalite ve benzetim konuları üzerinde yoğunlaşmaktadır. Prof. Baykasoğlu çok sayıda uluslararası dergide hakem ve yayın kurulu üyesi olarak görev yapmakta olup aynı zamanda Turkish Journal of Fuzzy Systems dergisinin eş-editörlüğünü yürütmektedir. Prof. Baykasoğlu'na 2007 yılında Türkiye Bilimler Akademisi Üstün Başarılı Genç Bilim İnsanı ödülü, 2008 yılında ODTÜ M. Parlar araştırma teşvik ödülü, 2010 yılında ise Tübitak Teşvik ödülü verilmiştir.