Gini Algoritmasını Kullanarak Karar Ağacı Oluşturmayı Sağlayan Bir Yazılımın Geliştirilmesi

M. Fatih ADAK[1], Nilüfer YURTAY[2]

[1,2] Bilgisayar Mühendisliği Bölümü, Sakarya Üniversitesi, Türkiye
fatihadak@sakarya.edu.tr, nyurtay@sakarya.edu.tr
(Geliş/Received: 03.01.2013; Kabul/Accepted: 23.12.2013)

Özet— Günümüzde veri madenciliğinin kullanımı yaygınlaşmış ve veri setleri devasa boyutlara ulaşmıştır. Bu veri setlerine veri madenciliği modellerini uygulamak ve detaylı analizler elde edebilmek için çeşitli araçlara ihtiyaç duyulmaktadır. Piyasada bu tür araçların bulunmasına rağmen kolay kullanım ve detaylı analizlere sahip araçların sayısı azdır ve bu araçlar genelde maliyeti fazla araçlardır. Dolayısıyla kolay kullanıma sahip veri madenciliği modellerini uygulayabilecek ücretsiz araçlara ihtiyaç duyulmaktadır. Bu çalışmada veri madenciliği modellerinden Gini algoritmasını veri seti üzerine uygulayıp karar ağacı oluşmasını sağlayan bir yazılım geliştirilmiştir.

Anahtar Kelimeler— gini algoritması, karar ağaçları, cart

Developing a Software Which Provides Creating Decision Trees by Using Gini Algorithm

Abstract— Today, the use of data mining become widespread and data sets has become very huge. Various tools are needed to Apply data mining models to these data sets and to get detailed analyses from those. Although there are similar tools available in the market, there aren't many tools which have easy to use and doing detailed analysis. Therefore there is a need for free software that has easy to use and can apply data mining models. In this study software that can apply Gini algorithm which is a data mining model to data sets and create decision trees has developed.

Keywords— gini algorithm, decision trees, cart

1. GİRİŞ (INTRODUCTION)

Veri madenciliği uygulamalarını gerçekleştirebilecek birçok ortam bulunmaktadır. Bu ortamlara örnek olarak SPSS Clementine, Rapidminer ve Weka yazılımları verilebilir. Bu yazılımlar çok detaylı analizler sunmasına rağmen birçoğu ücretli yazılımlardır. Ücretsiz olan yazılımlar ise ara yüzleri her kullanıcının kolay alışabileceği yazılımlar değildir. Bunların dışında küçük çapta yazılımlar bulunmaktadır. Fakat bunların birçoğunda birçok kısıtlama vardır ve çok dinamik yapılar değillerdir.

Literatürde bu çalışmadakine benzer düzeyde çalışmalar yapılmıştır. Berzal ve arkadaşlarının geliştirmiş oldukları Tminer yazılımı bir framework olup daha çok veri tabanı sistemleri üzerinde çalışmaktadır. Nesne ve model oluşturabilme özelliği bulunmaktadır [1]. Fdez ve arkadaşlarının geliştirmiş oldukları KEEL isimli yazılım veri madenciliği ve bulanık sistemler için bir araç niteliği taşımaktadır. Fdez çalışmasında birçok avantajından bahsetmiştir. Veri yönetimine tam anlamıyla izin veren ve her türlü veri seti ile çalışabilen bir yazılım geliştirmişlerdir [2]. Gorodetsky ve arkadaşları danışman tabanlı dağıtık veri madenciliği alanında kullanılabilecek bir yazılım aracı geliştirmişlerdir. Meta bilgiler ve kaynak tabanlı olarak çalışmaktadır. Gorodetsky'nin geliştirdiği yazılımın sınırlayıcı yanı sadece dağıtık (distributed) veri kaynakları üzerine çalışabiliyor oluşudur [3]. Bose ve arkadaşının geliştirmiş oldukları IDM ortamı, bu ortam danışman tabanlı veri madenciliği ortamı olup kullanıcı ara yüzü geliştirmişlerdir. Tasarladıkları kayıt ve mesaj panelleri ile danışmanların (agent) birbirleri ile haberleşmesi sağlanmıştır. Önerdikleri sistem bir karar destek sistemidir ve çalışmalarında bu sistemin avantaj ve limitlerinden bahsetmişlerdir. Limite bir örnek olarak kullanıcı tabanlı bir sistem olduğu için kullanıcının doğru amaçları girmesi beklenmektedir [4]. Java'da yazılmış bir

başka veri madenciliği yazılımı Bhat ve arkadaşları tarafından geliştirilmiş olup veri seti olarak sadece twitter üzerinde çalışmaktadır. Bhat ve arkadaşlarının yaptıkları çalışmada Dünya üzerinde seçilen bir bölge için gönderilen tweet dağılımını çıkarabilmektedirler [5].
Robu ve arkadaşları Weka yazılımı için bir dönüştürücü araç geliştirmişlerdir. Weka açık kaynak kodlu bir yazılım olduğundan araç geliştirip ekleme yapmak kolay olmatadır. Robu ve arkadaşlarının geliştirdikleri araç, Weka'nın okuyamayacağı veri tabanlarından Weka'nın okuyabileceği veri tabanlarına dönüşüm yaparak kullanıcının işini kolaylaştırmaktadır. MySql, MsSql gibi çeşitli veri tabanlarından okuma yapıp sağladığı ara yüzler ile veri üstünde istenen filtreler uygulanıp Weka'nın kullanabileceği veri seti tarzına dönüşüm yapılmaktadır [6].

Bu çalışmada geliştirilen yazılım sayesinde kullanıcı elindeki veri setini programa okutup Gini algoritmasını uygulayarak karar ağacı oluşturabilmektedir.

2. VERİ MADENCİLİĞİ (DATA MINING)

Veri madenciliği sayesinde elde bulunan veri üzerinde çeşitli algoritmalar uygulayıp veri hakkında farklı kararlara ulaşılabilir. Veri madenciliği çoğunlukla istatistiksel hesaplamalara dayanır. Sınıflama, kümeleme ve karar ağaçları oluşturma gibi çeşitli yöntemlere sahiptir.

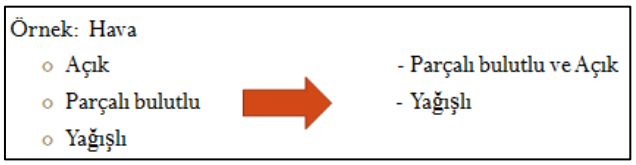
Veri madenciliği gözlemsel veri kümelerinin genelde büyük hacimli olanların analizini yapıp beklenmedik ilişkileri bulmak, yeni yöntemler kullanarak anlaşılabilir ve faydalı bir biçime sokmak olarak tanımlanabilir [6].
İki önemli veri analiz metodu olan sınıflandırma ve karar verme, önemli veri sınıflarını belirleyebilir veya yapıcı modeller oluşturabilirler.

Böylelikle gelecekteki veri hakkında tahmin ve yorum yapılabilmesini sağlayabileceklerdir. Sınıflandırma kategorik verilerin tahmininde kullanılır. Sınıflandırmada genellikle karar ağaçları, yapay sinir ağları ve genetik algoritmalar gibi teknikler kullanılır [7,8,9].

2.1 *Gini Algoritması*

Gini algoritması karar ağacı oluşturulmasında kullanılan bir algoritmadır. CART ağacı olarak tanımlanır.

CART (Sınıflandırma ve Karar Ağaçları) karar ağacı her bir karar düğümünden itibaren ağacın iki dala ayrılması ilkesine dayanır [10]. İlk hangi nitelikten bölüneceği ve bölünme değeri gini indeks değerine bakılarak hesaplanır. Gini indeks değeri veri setindeki varlıkların oranı olarak tanımlanabilir. İki varlığın gini değeri aynı çıkarsa sonuç dağılımları aynı demektir. Eğer veri setindeki bir nitelikte 3 veya daha fazla seçenek bulunuyorsa ve ikiden fazla bölünmeye izin verilmediği için birbirine yakın seçenekler gruplandırılır. Şekil 1'de bir örnek verilmiştir.



Şekil 1. Niteliğin bölünme örneği (The division sample of qualifications) [11]

Gini algoritması bütün veri setlerinde başarılı sonuç vermez ve bazı durumlarda ağaç sonlandırılamayabilir. Bu durumda aşağıdaki durma kurallarından biri veya bir kaçı uygulanabilir.

- Eğer düğüm saf hale gelmiş ise,
- Bütün düğümler saf hale gelmiş ise,
- Ağaç maksimum derinliğe ulaşmış ise,
- Minimum düğüm boyutuna ulaşılmış ise,

Bir niteliğin Gini değeri hesaplanmadan önce niteliğin Gini sol ve Gini sağ değerleri hesaplanmalıdır. Bu hesaplamalar Denklem 1 ve 2'deki gibi hesaplanır.

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

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

Denklem 1 ve 2 deki sembollerin sırasıyla açıklamaları;

k : Sınıfların sayısı
T : Bir düğümdeki örnek sayısı
Tsol : Sol koldaki örneklerin sayısı
Tsağ : Sağ koldaki örneklerin sayısı
Li : Sol kolda i kategorisindeki örneklerin sayısı
Ri : Sağ kolda i kategorisindeki örneklerin sayısı

Hesaplanan bu sol ve sağ değerler niteliğin Gini değerinin hesaplanmasında kullanılır. Bir niteliğin Gini değeri Denklem 3 kullanılarak bulunmaktadır.

$$Gini_j = \frac{1}{n}(|T_{sol}| Gini_{sol} + |T_{sağ}| Gini_{sağ}) \quad (3)$$

Her bir nitelik için hesaplanan Gini değerleri arasından en küçük olanı seçilir ve bölünme bu nitelik üzerinden gerçekleşir. Kalan veri seti üzerinde yukarıdaki bahsedilen adımlar tekrar uygulanır ve diğer bölünme hesaplanır.

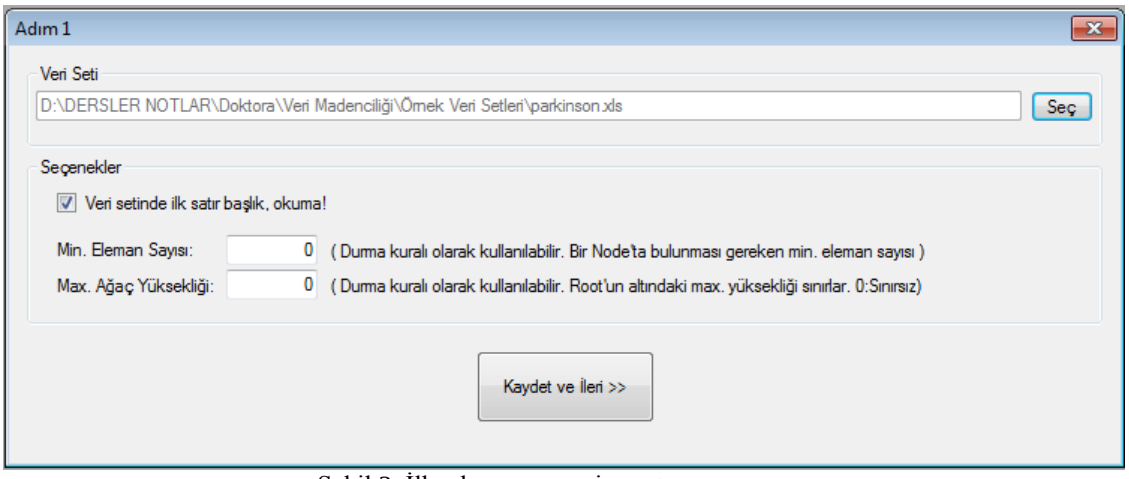
3. GELİŞTİRİLEN YAZILIM (DEVELOPED SOFTWARE)

Veri madenciliği algoritmalarından karar ağacı oluşturmaya yarayan Gini algoritmasını gerçekleştiren bir yazılım geliştirilmiştir. Yazılım Microsoft Visual C# 4.0 kullanılarak Windows uygulaması olarak yazılmıştır. Basit ve kullanışlı ara yüzü sayesinde birkaç adımda excel

dosyasından okunan veri seti durma kurallarının ve çıktı değerinin seçilmesi ile karar ağacını oluşturur.

3.1. *Verinin okunması (Reading of data)*

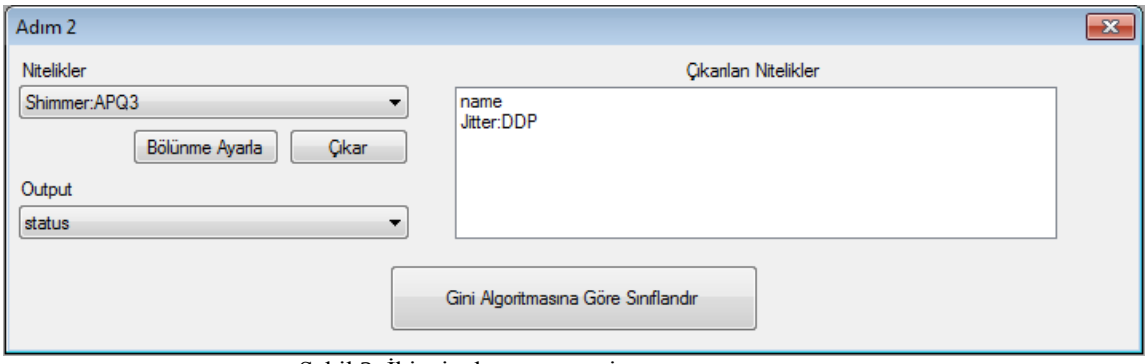
Geliştirilen program dosya olarak sadece excel dosyalarını okuyabilmektedir. Eldeki veri seti excel formatına dönüştürüldükten sonra programa rahatlıkla okutulabilir. Program ilk açıldığında Şekil 2'deki ekran çıkmaktadır.



Şekil 2. İlk adım penceresi (The first step window)

Seç butonuna tıklanıp veri seti seçildikten sonra istenirse seçenekler kısmı değiştirilebilir. Seçenekler kısmında eğer okutulan veri setinin ilk satırında başlıklar varsa bunu programa bildirilip okumaması sağlanabilir. Gini algoritması her veri setinde düğümü saf hale getirmeyebilir bu durumda karar ağacının bir durma koşulu olmalıdır. Bundan dolayı bu programa da iki durma koşulu eklenmiştir. Birincisi minimum eleman sayısı, bu koşul eğer düğüm belirtilen minimum eleman sayısına erişmiş ise veya bölündüğünde bu sayının altına düşecek ise bölünme durur. Maksimum Ağaç yüksekliği koşulunda ise girilen yükseklik değerine ağaç ulaştığında bölünme durur.

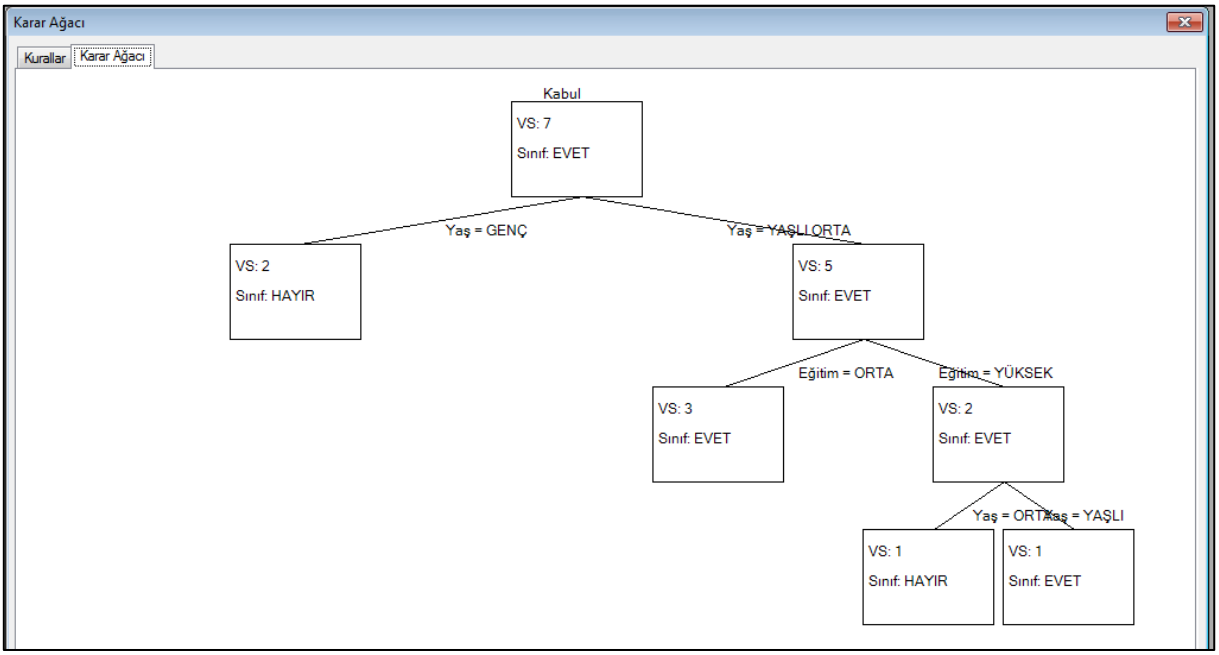
İlk adım geçildikten sonra ikinci adım penceresi ekrana gelir. Bu pencere Şekil 3'te görülmektedir.



Şekil 3. İkinci adım penceresi (The second step window)

Adım 2 penceresinde veri setinden okunan bütün nitelikler görülebilmektedir. Bu adımda istenirse bazı nitelikler oluşturulacak karar ağacından çıkarılabilir ve nitelikler sayısal değerler değilse bu durumda istendiğinde bölünme ayarlanabilir. Bölünmeden kasıt eğer bir nitelik 3 veya daha fazla farklı durum içeriyorsa bölünmenin nasıl olması gerektiğini belirlemeyi sağlar. Örneğin yaş niteliği için veri setinden okunma sırası (orta, genç, yaşlı) ise ve bölünme belirtilmez ise ikili dallanma [orta], [genç, yaşlı] şeklinde olabilecektir ki bu istenen bir durum değildir. Dolayısıyla bölünme ayarla butonuna basılıp sıra (genç, orta, yaşlı) şeklinde değiştirilebilir. Veri setindeki niteliklerden biri Gini algoritması sağlıklı hesaplanabilmesi için sonuç niteliği olarak seçilmelidir. Sonuç niteliği seçildikten sonra Gini algoritmasına göre sınıflandır butonuna basılır. Bu aşamadan sonra program belirlenen kriterlere göre algoritmayı uygulayacak ve karar ağacını oluşturacaktır.

Kurallar ve karar ağacı iki ayrı panelde gösterilmektedir. Örneğin Tablo 1'deki veri seti programa okutulmuş bölünmeler ayarlanmış ve Şekil 4'teki gibi karar ağacı elde edilmiştir.



Şekil 4. Okutulan örnek veri setinden oluşan karar ağacı (A decision tree consisting of instructed sample data sets)

Tablo 1. Örnek veri seti [11] (Example data set)

Eğitim	Yaş	Kabul	Cinsiyet
Orta	Yaşlı	Evet	Erkek
İlk	Genç	Hayır	Erkek
Yüksek	Orta	Hayır	Kadın
Orta	Orta	Evet	Erkek
İlk	Orta	Evet	Erkek
Yüksek	Yaşlı	Evet	Kadın
İlk	Genç	Hayır	Kadın

Şekil 4'te görülen kurallar sekmesine tıklayıp, kurallar alt alta da görülebilir. Bahsedilen bu kurallar Şekil 5'te görülmektedir.

Şekil 5. Okutulan örnek veri setinden oluşan kurallar (Rules consisting of instructed sample data sets)

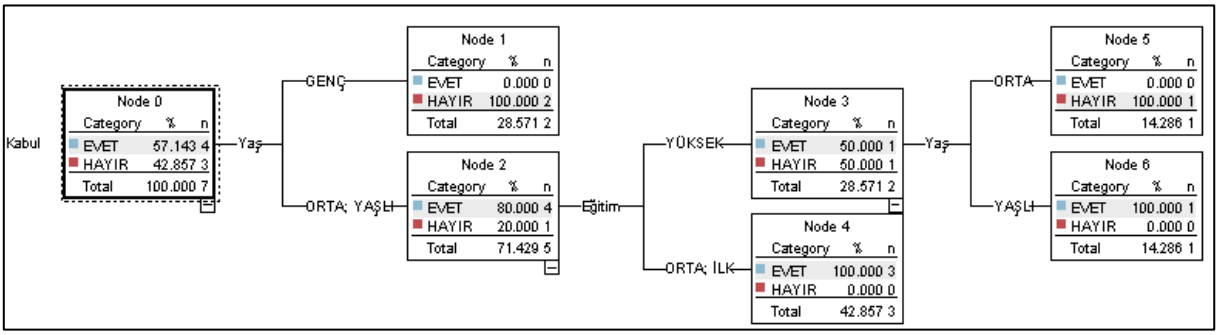
Oluşan karar ağacı istendiği durumda resim olarak ta kaydedilebilmektedir.

4. BULGULAR (FINDINGS)

Geliştirilen yazılımın doğru sonuç verdiğini test etmek için uluslararası kabul görmüş SPSS Clementine 12 programının verdiği sonuçlar ile karşılaştırılmıştır. İki örnek veri seti üzerinde yapılan testler göstermiştir ki bu çalışmada geliştirilen yazılım doğru sonuç üretmektedir.

4.1. *İlk örnek, basit bir veri seti (First example, A simple data set)*

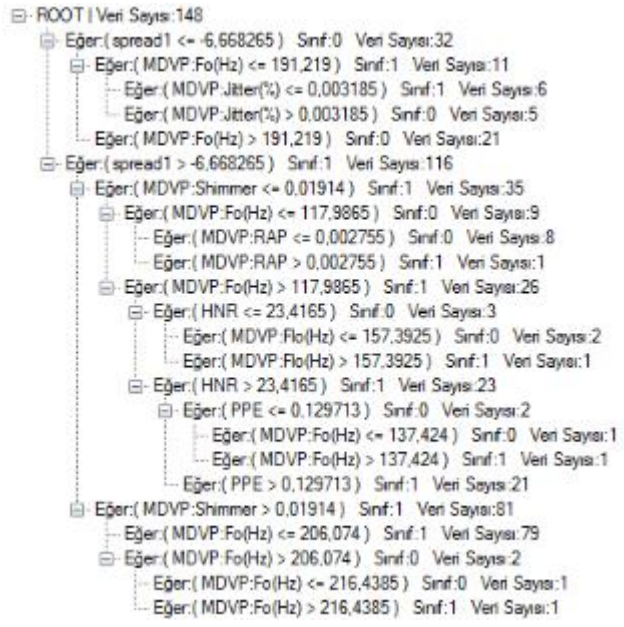
Tablo 1'de gösterilen bu veri seti eğitim, yaş, cinsiyet ve kabul niteliklerinden oluşmaktadır. Çıktı niteliği olarak kabul seçilmiş ve karar ağacı oluşturulmuştur. Aynı veri seti Clementine programı kullanılarak karar ağacı oluşturulduğunda Şekil 6'daki gibi bir karar ağacı elde edilmektedir. Şekil 4 ve Şekil 6 karşılaştırıldığında karar ağaçlarındaki dallanmaların aynı olduğu görülecektir.



Şekil 6. Clementine programının oluşturduğu karar ağacı (Decision tree created by clementine program)

4.2. *İkinci örnek parkinson veri seti (Second example, Parkinson data set)*

Veri setleri barındıran bir web sitesinden alınmış olan Parkinson hastalığı veri seti %70'i eğitim, %30'u test için ayrılmıştır [12]. Bu çalışmada geliştirilen yazılım kullanılarak veri setinin %70'den karar ağacı oluşturulmuştur. Oluşan bu karar ağacı Şekil 7'de görülmektedir. %30 test verisi 47 adet veri yapmaktadır. 47 adet veri ağaç üzerine uygulanıp bu verilerin karar ağacında verdikleri sonuçlar kaydedilmiştir. Aynı zamanda bu 47 adet verinin gerçek sonucu da alınıp programın Roc(Receiver Operating Characteric) analizi yapması sağlanmıştır [13,14]. Roc analizinden elde edilen sonuçlar Tablo 2'de görülmektedir.



Şekil 7. Geliştirilen yazılımın oluşturduğu karar ağacı
(A decision tree created by developed software)

DP: Gerçek durum pozitifken test sonucu da pozitif çıkan durumlar
YN: Gerçek durum pozitifken test sonucu negatif çıkan durumlar
YP: Gerçek durum negatifken test sonucu pozitif çıkan durumlar
DN: Gerçek durum negatifken test sonucu da negatif çıkan durumlar

Tablo 2. Roc analizi parametreleri (Roc analysis parameter)

	Gerçek Durum		
Test Sonucu	Pozitif	Negatif	Toplam
Pozitif	32 (DP)	3 (YP)	35
Negatif	5 (YN)	7 (DN)	12
Toplam	**37**	**10**	**47**

Doğruluk(Accuracy): (DP + DN)/(DP+DN+YN+YP)
= 39 / 47 = **0,83**

Duyarlılık (Sensitivity): *DP/(DP+YN)*
= 32/37 = **0,86**

Seçicilik (Specificity): *DN/(YP+DN)*
= 7/10 = **0,7**

Buradaki %86'lık duyarlılık değeri, testin, gerçek pozitif durumlar içinden pozitif olan durumları ayırma yeteneğidir. Testin, gerçek negatif durumlar içinden negatif olan durumları ayırma yeteneği de %70 olarak elde edilmiştir.

5. SONUÇLAR (RESULTS)

Bu çalışmada geliştirilen yazılım Gini algoritmasını kullanarak yüklenmiş olan veri setinden kurallar ve karar ağacı oluşturabilmektedir. Kabul görmüş Clementine 12 programı ile geliştirilen yazılım iki farklı veri setinde karşılaştırılmış ve sonuçların aynı çıktığı gözlemlenmiştir. Seçilmiş olan veri setlerinden biri niceliksel diğeri sayısal olduğu için farklı durumlarda da doğru sonuç ürettiği görülmüştür.
Bu çalışmada geliştirilen yazılım sayesinde oluşan karar ağacı üzerinde test işlemi uygulanabilmekte ve Roc analizi çıkarılabilmektedir. Bu sonuçlara kullanıcı basit birkaç adımda ulaşabilmektedir.

KAYNAKLAR (REFERENCES)

[1] F. Berzal, J. C. Cubero, Jimenez A., **The design and use of the TMiner component-based data mining framework**, Expert Systems with Applications, 36, 7882-7887, 2009.

[2] J. A. Fdez, S. Garcia, Berlanga F. J., KEEL: A data mining software tool integrating genetic fuzzy systems, **3rd International Workshop on Genetic and Evolving Fuzzy Systems**,84-88 ,March, 2008.

[3] V Gorodetsky., O. Karsaeyv, Samoilov V., Software tool for agent-based distributed data mining, **KIMAS 2003 Boston**, USA.

[4] R. Bose, V. Sugumaran, IDM: an intelligent software agent based data mining environment, **International Conference on Systems Man and Cybernetics**,2888-2893 ,11-14 Oct. 1998.

[5] F. Bhat, M. Oussalah, K. Challis, T. Schnier, A software system for data mining with Twitter, **10th IEEE International Conference On Cybernetic Intelligent Systems**, 139-144 ,1-2 Sept. London, UK, 2011.

[6] R. Robu, V. S. Tivadar, Arff convertor tool for WEKA data mining software, **International Joint Conferences on Computational Cybernetics and Technical Informatics**, 247-251,27-29 May. Timisora, Romania, 2010.

[6] D. Hand, H. Mannila, P. Smyth, **Principles of data mining**, 1st ed., A Bradford Book The MIT Press, London, 2001.

[7] A. Berson, S. Smith, K. Thearling, "**Building Data Mining Applications for CRM**", McGraw-Hill Professional Publishing, New York, USA, (2000).

[8] S. Chaudhuri, **Data Mining and Database Systems : Where is the Intersection?**, IEEE Bulletin of the Technical Committee on Data Engineering, 21 (1) (1998) 4 - 8.

[9] S. Özekeş, A. Y. Çamurcu, **Classification and prediction in a data mining application**, Journal of Marmara for Pure and Applied Sciences, 18/159-174, 2002.

[10] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, **Classification and Regression Trees**, Wadsworth, Belmont, 1984.

[11] Y. Özkan, **Veri Madenciliği Yöntemleri**, 1st ed., Papatya yayıncılık, Türkiye, 2008.

[12] M. A. Little, P.E. McSharry, S. J. Roberts, D. A. E. Costello, I. M. Moroz, '**Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection**',. BioMedical Engineering OnLine 2007, 6:23 (26 June 2007).

[13] A. Dirican, Evaluation of the diagnostic test's performance and their comparisons**. Cerrahpaşa J Med ,** 32 (1): 25-30, 2001.

[14] L. Tomak, Y. Bek, İşlem Karakteristik Eğrisi Analizi Ve Eğri Altında Kalan Alanların Karşılaştırılması, **Journal of Experimental and Clinical Medicine**, Vol:27, no:2, s:58-65, 2010.