Sınıflandırıcı Topluluklarının Dengesiz Veri Kümeleri Üzerindeki Performans Analizleri

Faruk BULUT

İzmir Katip Çelebi Üniversitesi, Bilgisayar Mühendisliği, İzmir, Türkiye
faruk.bulut@ikc.edu.tr
(Geliş/Received: 16.12.2015; Kabul/Accepted: 16.05.2016)
DOI: 10.17671/btd.81137

Özet— Kolektif öğreniciler diğer bir adıyla sınıflandırıcı toplulukları, sınıflandırmadaki doğruluk oranını artırmak için kullanılan yeni ve yaygın bir yapay öğrenme yöntemidir. Literatürde önerilmiş birçok kolektif öğrenme metodu vardır. Bu öğreniciler önceden hazırlanmış olan etiketli veri setleri üzerinde eğitilerek bir sınıflandırma modeli oluştururlar. Veri setleri içerisinde bulunan her bir sınıf türüne ait örnek sayıları her zaman dengeli olamayabilir. Ayrıca dengesiz sınıf dağılımların bulunduğu veri setleri üzerinde sınıflandırıcılar istenilen düzeyde performans gösteremeyebilir. Bu çalışmada, yüksek başarı göstermesi beklenen öğrenici topluluklarının dengesiz sınıf dağılımlarına sahip veri setleri üzerindeki sınıflandırma başarıları analiz edilmeye çalışılmıştır. Yapılan deneysel uygulamalarda Karar Ağaçları, Bagging, Boosting, Random Subspaces, Dagging ve Decorate yöntemleri belirlenen veri setleri üzerinde denenmiştir. Yapılan istatistiksel analizlerde en başarılı sınıflandırıcı topluluğunun Boosting yöntemi ile elde edildiği görülmüştür. Ayrıca kolektif öğrenicilerin başarı ve başarısızlıklarının nedenleri üzerinde bazı değerlendirmeler yapılmıştır.

Anahtar Kelimeler **—** Kolektif öğreniciler, dengesiz veri setleri, performans analizi.

Performance Analysis of Ensemble Methods on Imbalanced Datasets

Abstract**—** Ensemble learners, in other words collective classifiers are commonly used in order to increase the classification accuracy. Many collective learning methods in the literature have been proposed in recent years. A classification model is created by these learners whom have been trained on the predefined labeled datasets. The distribution of class samples in a dataset may not be always distributed evenly. In addition, the classifiers may not give an accurate classification results over those types of datasets. In the present study, the performance analysis of some ensemble methods over imbalanced datasets has been performed. In the experimental applications, the ensemble methods of Bagging, Boosting, Random Subspaces, Dagging and Decorate methods have been applied to the selected benchmark datasets. In the statistical part, the most successful ensemble method has been Boosting. Additionally, some assessments over the failure and success of the ensemble methods have been done.

Keywords **—** Ensemble methods, imbalanced datasets, performance analysis.

1. GİRİŞ (INTRODUCTION)

Sınıflandırma yöntemleri makine öğrenmesi, şekil tanıma ve veri madenciliği alanlarında kullanılmaktadır. Literatürde öne sürülmüş birçok yöntem vardır. Karar ağaçları, destek vektör makineleri, çok katmanlı algılayıcılar, en yakın komşuluk algoritmaları en çok kullanılan ve bilinenleridir. Bu tür algoritmalar tekil öğreniciler (*base learner*) olarak adlandırılır ve belirli bir veri seti ile eğitilerek sınıflandırmama aşamasında kullanılırlar.

Son yıllarda tekil öğrenicilerden oluşturulmuş değişik sınıflandırıcı toplulukları bilimsel çalışmalarda öne sürülmüştür. Kolektif öğreniciler olarak da adlandırılan sınıflandırıcı toplulukları, sınıflandırma aşamasında tekil öğrenicilere göre daha yüksek bir başarı oranı sundukları için tercih edilmektedir ve geniş bir kullanım alanına sahiptirler. Kolektif yöntemlerin temel mantığında sınıflandırıcıların kararlarının birleştirilmesi ve komite kararının oluşturulması vardır.

Belirli bir veri seti üzerinde işlem yapan sınıflandırıcı topluluğunda kimi zaman birbirinden farklı sınıflandırıcı yöntemleri kullanıldığı gibi kimi zaman da aynı

sınıflandırıcı farklı parametrelerle de eğitilebilir. Bazen aynı eğitim setinin bazı örnekleri seçilerek yeni eğitim setleri oluşturulur bazen de aynı eğitim setinin bazı öznitelikleri seçilerek yeni eğitim setleri oluşturulur.

Gerçek hayatta bazı problemlerin çözümü için yapay karar destek sistemlerinden faydalanılmak istenir. Bu durumda sınıflandırıcılar kullanılır. Sınıflandırıcı mekanizmaları için hazırlanan veriler dengeli olmayabilir. Bir veri setinde her sınıftan hemen hemen aynı sayıda örneğin bulunması yani dengeli bir dağılımın sağlanmış olması beklenen bir durumdur. Fakat bunun aksinin olduğu durumlarda yani dengesiz veri kümelerinde (*imbalanced or skewed dataset*), sınıflandırıcılar eğitilirken örnek sayısının çoğunlukta olduğu sınıf kümesine doğru bir eğilim gösterir. Diğer bir deyişle sınıflandırıcıda örnek sayısı fazla olan sınıf etiketi ile eğitileceği için sistemde bir önyargı (*bias*) oluşmaktadır. Ayrıca azınlıkta olan sınıf etiketi ile sınıflandırıcı kendini yeterince eğitemediği için ilerleyen aşamalarda başarılı sınıflandırma yapamamaktadır. Esasen bu istenmeyen bir durumdur. Bu gibi durumlarda sınıflandırıcının her bir sınıf etiketi için yüksek başarı göstermesi beklenmektedir.

Literatürde dengesiz veri setleri üzerinde yapılan bazı çalışmalar ve önerilen değişik yöntemler vardır. Bir çalışmada dengesiz veri kümeleri için en uygun öğrenme yönteminin nasıl olması gerektiği önerilmiştir [1]. Bu çalışmada klasik sınıflandırıcı yöntemlerin doğal olarak dağılımın fazla olduğu sınıf türlerine doğru yöneldiği vurgulanmış ve bu problemin giderilmesi için bilinen yöntemler birleştirilerek kullanılmıştır.

Bu alanda yapılmış kapsamlı bir araştırma "*Learning from Imbalanced Data*" isimli bir makalede yer almaktadır [2]. Bu çalışmada dengesiz verilerin hangi yaşamsal alanlarda bulunduğu incelenmiş, bilgi keşfinin bu gibi durumlarda nasıl yapılması gerektiği tespit edilmiş ve öğrenme işleminin aşamaları ile ilgili öneriler sunulmuştur. Ayrıca performans analiz yöntemlerinin neler olabileceğine yer verilmiştir.

Konu ile alakalı bir başka çalışmada Borderline-SMOTE yöntemi önerilmiştir [3]. Bilinen SMOTE (*Synthetic Minority Oversampling Technique*) yönteminde dengesiz veri dağılımının bulunduğu sınıf türü yapay olarak çoğaltılır ve dengelenme bu sayede gerçekleştirilir. Borderline-SMOTE metodunda ise azınlıkta bulunan sınıf türüne ait örneklerin oluşturduğu kümelerin sınır çizgilerinde bulunan örnekler SMOTE ile çoklanmıştır. Yaptıkları deneysel uygulamada Borderline-SMOTE yönteminin daha yüksek başarı sağladığı belirtilmiştir.

Destek vektör makinelerinin (SVM) bu tür veri setlerine nasıl uygulanması gerektiği bir başka çalışmada incelenmiştir [4]. Bazı hazır ve ortak kullanıma açık veri setleri üzerinde denemeler yapılmıştır. Dağılımı diğer sınıf türlerine göre az olan örnekler çoğaltma (*resampling*) yöntemiyle dengeli bir dağılım gerçekleştirilmeye çalışılmıştır. Ayrıca kullanılan SVM yöntemi SMOTE algoritmasıyla kıyaslanmıştır.

Dengesiz veri kümelerinde sınıf türlerine ait sınıf çizgilerinin nasıl belirleneceği ile ilgili öneriler başka bir çalışmada sunulmuştur [5]. Bu çalışmanın asıl amacı SVM sınıflandırıcısının en yüksek performansta çalışabilmesi için bu tür veri setlerinin nasıl ele alınması gerektiği konusunda bazı ölçütlerin belirlenmesini olmuştur.

Bu alanda yapılan son çalışmalardan birinde [6] veri setinde bulunan azınlıktaki örneklerin sayısının artırılması ile ilgili RUS (*Random Under-Sampling*) gibi değişik yöntemler incelenmiştir. Örneklerin çoğaltılması ile veri setinde elde edilmeye çalışılan dengenin ne düzeyde olacağı ile ilgili bir de otonom bir sistem önerilmiştir. Ayrıca DCS (*Contribution Sampling Method*) ismi verilen bir yöntem ile yapılan örnek çoğaltma işleminde SVM sınıflandırıcısının performansı ROC (*Receiver Operating Characteristic*) eğrileri ile test edilmiş ve kıyaslanmıştır.

Ayrıca yayınlanan başka bir makalede [7] örnek tabanlı bir sınıflandırıcı olan *k*-NN algoritması ile oluşturulan AdaBoost yönteminin, ikiden fazla sınıf türü barındıran dengesiz veri setleri üzerinde nasıl kullanılacağı ile ilgili bir çalışma gerçekleştirilmiştir. Yapılan başarım analizleri 19 veri seti üzerinde ROC eğrisi altında kalan alanın (AUC) hesaplanıp karşılaştırılması ile yapılmıştır. Denetimli öznitelik seçim yöntemlerinden biri olan BPSO (*Base Particle Swarm Optimization*) modelinin kullanıldığı çalışmada AUC ölçütü de uygunluk fonksiyonu olarak tercih edilmiştir.

2016 yılı içerisinde yayımlanan başka bir çalışmada çok sınıflı dengesiz veri kümeleri için lokal özellikleri baz alan adaptif bir model önerilmiştir [8]. AMCS (*Adaptive Multiple Classifier System*) ismini verdikleri modelde hemen hemen her türdeki dengesiz veri setleri için başarılı bir sınıflandırma yapabildikleri belirtilmiştir. Dengesiz veri setlerinin kategorilere ayırırken örnek sayıları, sınıf etiketi türü, örnek sayısı gibi temel istatistiksel veriler temel alınmıştır. Her bir veri seti türü için öznitelik seçimi, örnekleri çoğaltma ve kolektif metot seçimi gibi üç bileşenin temel alındığı belirtilmiştir. AMCS'nin deneysel uygulamalarında test amaçlı seçilen her bir veri seti için bu üç bileşenin değişiklik göstereceği vurgulanmıştır. Bilinen beş kolektif sınıflandırıcısı ile yapılan performans karşılaştırmalarında önerilen AMCS yöntemin daha başarılı sonuçlar verdiği yazılmıştır.

Çalışmamızdaki amaç kolektif öğrenici topluluklarının dengesiz veri kümeleri üzerindeki sınıflandırma başarılarını incelemek ve en başarılı yöntemin belirlenmesini sağlamaktır. Çalışmamızın geriye kalan kısmı üç ana bölümden oluşmaktadır. İkinci bölümde komite sınıflandırıcılarının açıklanmasına, üçüncü bölümde dengesiz veri setleri için kullanılabilecek en uygun performans ölçütüne, dördüncü bölümde ise elde edilen deneysel sonuçların incelenmesine ve son bölümde de yorumlara yer verilmiştir.

2. KOLEKTİF SINIFLANDIRICILAR (ENSEMBLE CLASSIFIERS)

Her bir kolektif sınıflandırıcı tekniği tekil bir sınıflandırıcı kullanmak zorundadır. Bilindiği üzere tekil sınıflandırıcı olarak yaygın bir kullanım alanına sahip karar ağaçları, destek vektör makineleri, Naive Bayes metodu, lineer ayırıcılar ve yapay sinir ağları gibi algoritmalar mevcuttur. Hızlı eğitilebilmeleri, çabuk sınıflandırma yapabilmeleri, doğruluk oranlarının yüksek olması ve şeffaf bir yapıda (*white box*) olmaları nedeniyle karar ağaçları kolektif yöntemlerde tekil öğrenici olarak kullanılırlar [9]. Çalışmamızda, aynı tekil sınıflandırıcı tüm kolektif

öğrenicilerde kullanılarak kıyaslama yapılabilmesi mümkün olmuştur.

2.1.Karar Ağacı (Decision Tree)

Sınıflandırma ağaçları sorgu noktasını bir sınıfa yerleştirmeyi amaçlar. Karar ağacı yönteminde, bir ağaç yapısı oluşturulur. Ağacın yaprakları üzerinde sınıf etiketleri ve gövdeden yapraklara giden çizgiler üzerinde de işlemeler yer almaktadır. Ağacın oluşturulması sırasında, üzerinde eğitim yapılan veri setinde her bir sınıf etiketinin entropi değerlerine dayalı bölümleme yapılır. Bu işlem, özyinelemeli bir şekilde tekrarlanır ve tekrarlama işlemi önemini yitirene kadar sürer. ID3 ve C4.5 en bilinen yöntemlerdendir. CART (*Classification And Regression Tree*) ise hem sınıflandırma hem de regresyon işlemi yapan karar ağacının kısaltılmış adıdır.

2.2.Bagging

Bagging (*Bootstrapping Aggregation*) metodu, var olan bir eğitim setinden yeni eğitim setleri türeterek temel öğreniciyi yeniden eğiten bir yöntemdir. Bagging'de amaç yeni veri setleri türeterek farklılıkları oluşturmak ve bu sayede toplam sınıflandırma başarısını artırmaktır. Eğitim setinden yaklaşık olarak %63,2 kadar orijinal örnek rastgele alınır ve alınan örneklerden bazıları çoğaltılarak eğitim seti %100'e tamamlanır. Bu yöntemle birbirinden farklı bir miktar eğitim seti elde edilir. Her eğitim seti aynı temel öğreniciye uygulanır ve alınan kararlar ağırlıklı oylama yöntemiyle birleştirilir [10]. Eğitim setinden %63.2 kadar örnek seçilmesindeki neden şu formülle açıklanır:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} = 0{,}368 \qquad (1)$$

Burada *n* işlem sayısını veya eleman sayısını gösterir. *n* değeri sonsuza giderken doğal logaritma tabanı olan e sayısının tersi elde edilmiş olur.$(1-0{,}368=0.632)$

N sonsuza giderken Bu yöntemle orijinal eğitim kümesinden yaklaşık olarak %36'sı büyük ihtimalle hiçbir zaman seçilmemiş olacaktır. Bagging yönteminde seçilmeyenlerin seçilmesini sağlamak için eğitim setinden %63,2 kadar örnek rastgele seçilir ve yeni bir eğitim seti oluşturulur.

2.3. AdaBoost

AdaBoost en çok kullanılan ardışık topluluklarla öğrenme (*Boosting*)yöntemidir ve ilk olarak Freund ve Schapire tarafından önerilmiştir [11]. Diğer yöntemlerle kıyaslandığında tahmin hızının yüksek olması, daha az hafıza kullanması, uygulanabilir olması gibi özelliklerinden dolayı tercih edilmektedir.

Öğrenicilerin eğitileceği örneklerin seçiminde önceki temel öğrenicilerin hata yaptıkları örneklere öncelik verilmektedir. Diğer bir kolektif öğrenme metodu olan Bagging yönteminde her bir iterasyonda tüm örneklerin eğitim kümesine seçilme şansları ve olasılıkları aynıdır. Fakat Boosting'de her iterasyonda örneklere ait seçilme olasılıkları güncellenmektedir. Bu da doğru verilen kararlardan çok, yanlış verilen kararlar üzerine odaklanılmasını sağlayan bir yöntem olmasını sağlamaktadır [12]. Bu sayede sistemin daha doğru tahmin yapması sağlanmış olmaktadır.

AdaBoost algoritması, zayıf sınıflandırıcılardan oluşan bir topluluk ile çalışır. Zayıf sınıflandırıcılar veri setindeki her bir öznitelikten oluşturulur. Komitedeki bu sınıflandırıcıların karar sınırları her bir öznitelik için pozitif ve negatif örneklerin ağırlıklı ortalaması ile bulunur. Hata oranı en düşük olan zayıf sınıflandırıcılar kullanılarak güçlü bir sınıflandırıcı oluşturur. Bu sayede güçlü sınıflandırıcı içerisinde yer almayan zayıf sınıflandırıcılara ilişkin öznitelikler elenmiş olur. AdaBoost yönteminin kaba kodu (*pseudo code*) şu şekildedir [13]:

Adım 1: Veri setindeki *n* adet eğitim örnek $\{(x_1,y_1), \ldots ,(x_n,y_n)\}$ şeklinde verilmiş olsun. $y_i \in \{-1,+1\}, x_i \in X$ tanımlamasında x_i, her bir örneğin sınıf etiketi; y_i ise regresyon algoritmasının verdiği karardır. Pozitif örnekler için $y_i = +1$, negatifler için $y_i = -1$ olsun.

Adım 2:a pozitif örneklerin, *b* de negatif örneklerin sayısı olmak üzere *n=a+b*'dir. Ağırlıklar $w_{1,i} = \frac{1}{2b}, \frac{1}{2l}$ olacak şekilde her $y_i \in \{0,+1\}$ için ilklenir.

Adım 3: İ çevrim sayısı olmak üzere, her bir t=1, … ,*I* için:

a) Ağırlıklar normalize edilir: $w_{1,i} \leftarrow \frac{w_{1,i}}{\sum_{j=1}^{n} t,j}$

b) Veri setinde bulunan her bir *j* özniteliği için, bu *j* özniteliğini kullanan h_j sınıflandırıcısı eğitilir. w_t ağırlığına bağlı olarak hata oranı şu şekilde bulunur:$\varepsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$

c) En az ε_j hatasına sahip h_t sınıflandırıcısı seçilir.

d) Ağırlıklar güncellenir: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$

Burada x_i doğru olarak sınıflandırma yaptıysa $e_i = 0$, aksi halde $e_i = 1$ olur. $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ olarak hesaplanır.

Adım 4: $\alpha_t = \log\frac{1}{1-\beta_t}$ olarak alındığında $h(x)$ sınıflandırıcısının son durumu şu şekilde olur:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^{I} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{I} \alpha_t \\ 0, & diğer\ durmlar \end{cases} \qquad (2)$$

AdaBoost yönteminde, en güçlü zayıf sınıflandırıcıların bir araya getirilmesiyle güçlü bir sınıflandırıcının oluşturulması amaçlanmaktadır. Bunun için bu teknikte işlemler her bir eğitim örneği için eşit bir *D* dağılımıyla başlar. Her çevrimde sınıflama performansına bağlı olarak en iyi zayıf sınıflandırıcı tespit edilir ve ağırlıklar güncellenir. Güncellenen değerler ile bir olasılık dağılım fonksiyonu elde edilir. İlerleyen adımlarda da bu işlemler tekrarlanır. Belirlenmiş sayıdaki işlem sonucunda en güçlü zayıf sınıflandırıcıların birleştirilmesiyle yüksek performanslı bir sınıflandırıcı elde edilmiş olur.

2.4. Dagging

Dagging yöntemi ilk olarak Ting ve Witten tarafından önerilmiştir [14] ve Bagging yöntemi ile benzerlikler taşımaktadır. Bagging'de çoklama işlemi varken burada ayrık alt veri kümeleri oluşturma vardır. Bu meta sınıflandırıcı öncelikle veri setini belirli bir sayıda ayrık bölümlere ayırır. Her bir ayrık alt küme belirli veri öğrenici ile eğitilir. Sistemin verdiği karar ise alt kümelerin verdikleri kararların ortalamasının alınması ile yapılır. Ebetteki bu yöntem yoğun bir veri setinde iyi bir sınıflandırma performansı sergiler.

2.5. Decorate

Melville ve Mooney tarafından önerilen başka bir yöntem ise Decorate meta algoritmasındır [15]. Bu yöntemde aynı veri seti üzerinde farklı kararlar verebilecek bir birinden farklı sınıflandırıcıların inşa edilmesi vardır. Temel sınıflandırıcı olarak karar ağaçları tercih edilmektedir. Burada kolektif sınıflandırıcılar topluluğunda farklılıkların (*diversity*) oluşturularak kümülatif sınıflandırma başarısının artırılması için olasılık tabanlı yapay yeni veri setleri türetilmektedir. Her bir veri seti farklılığı oluşturmaktadır ve toplam karar ağırlıklı oylama (*majority voting*) ile bulunmaktadır.

2.6.Rastsal altuzaylar (Random SubSpaces)

Rastsal altuzaylar, Rastsal Ormanlar tekniğinin genelleştirilmiş bir türüdür. Rastgele ormanda temel sınıflandırıcı olarak karar ağaçlarının kullanıldığı bilinmektedir. Rastsal Altuzaylarda ise temel sınıflandırıcı olarak farklı sınıflayıcılar seçilebilmektedir. Rastsal altuzaylar yönteminde, belirli bir öğrenicinin eğitim setinin farklı öznitelikleriyle eğitilmesiyle sınıflandırma başarısını artırması hedeflenir. Burada eğitim setinin boyutları azaltılarak yeni eğitim setleri oluşturulur. *M* özniteliği bulunan bir veri setinden rastgele *N* adet özniteliği bulunan (*N*<*M*) yeni veri setleri türetilir. Yani eğitim setinde bulunan bazı öznitelikler silinip yeni eğitim setleri türetilir. Daha sonra aynı temel öğrenici ile bu eğitim setleri üzerindeki sınıflandırma başarısı hesaplanır. Öğrenicinin türetilen veri setleri üzerindeki kararları birleştirilerek ortak sınıflandırma kararı oluşturulur [16].

3. PERFORMANS ÖLÇÜM KRİTERLERİ (PERFORMANCE MEASUREMENT CRITERION)

3.1.Çapraz Geçerleme (Confusion Matrix)

Sınıflandırıcıların bir veri seti üzerindeki sınıflandırma performansı, çapraz geçerleme işlemiyle test edilir. Literatürde çeşitli çapraz geçerleme yöntemleri vardır. LOO, 10-Fold ve 5x2 çapraz geçerleme bunlardan bazılarıdır. Çalışmamızda 10-Fold çapraz geçerleme kullanılmıştır. Bu yöntemde veri seti 10 parçaya bölünür, bir parçası test seti kalan dokuz parçası da eğitim seti olur. Sınıflandırma işlemi bu bölümlemeye göre yapıldıktan sonra sırasıyla kalan onluk parçalar test seti olur. Her bir parçanın doğruluk oranı hesaplanır ve genel ortama alındıktan sonra sınıflandırma başarısı bulunmuş olur.

3.2. Karmaşıklık Matrisi (Confusion Matrix)

Sınıflandırma modellerinin başarımları ile ilgili değerlendirmelerin yapılabilmesi için Karmaşıklık Matrisi kullanılmaktadır. "Hata Matrisi" olarak da isimlendirilebilecek bu yöntem bu iki sınıflı bir veri seti ile alakalı belirli bir sınıflandırıcının verdiği sonuç şu şekilde olur:

Tablo 1. Karmaşıklık Matrisi
(Confusion Matrix)

		Tahmin Edilen Sınıf	
		C_1	C_2
Gerçek Sınıf	C_1	TP	FN
	C_2	FP	TN

C_1 ve C_2 sınıf türleri olmak üzere TP (*True Positive*) doğru sınıflandırılmış pozitif örnek sayısını, TN (*True Negative*) doğru sınıflandırılmış negatif örnek sayısını, FP (*False Positive*) yanlış sınıflandırılmış pozitif örnek sayısını, FN (*False Negative*) de yanlış sınıflandırılmış negatif örnek sayısını göstermektedir. Doğruluk (*Accuracy*), Hata (*Error*), Netlik (*Presicion*), Hassasiyet (*Recall*), F1-Score ve MCC gibi başarı ölçütleri bu tablodan yararlanılarak bulunmaktadır.

3.3.Sıfır Kuralı (Zero Rule)

İki sınıflı verilerde sınıflandırma başarısı doğruluk (*Accuracy*) değerine bakarak değerlendirilmesi büyük bir hatadır. Örneğin iki sınıflı bir veri setinde bir sınıf etiketinin bulunma olasılığı 0.95 oranında ise rastgele sınıflandırma başarısı (Zero Rule) en az %95 olmalıdır. Her hangi bir algoritma, rastgele sınıflandırma işlemi olan Zero Rule değerinden daha yüksek sonuçlar vermelidir. Bu nedenle dengesiz veri dağılımların olduğu veri setlerinde doğruluk başarı oranı için MCC ölçütü daha uygundur.

3.4. MCC

MCC (*Matthews Correlation Coefficient*), sınıf dağılımlarının dengesiz olduğu durumlarda bile en iyi sonucu vermektedir. MCC'de çıktı değeri -1 ile +1 arasında değişmektedir. 0 değeri rastgele sınıflandırma durumunu, -1 değeri var olan gerçek değerler ile sınıflandırıcının verdiği kararların tamamen birbirinden zıt olduğunu göstermektedir. +1 ise sınıflandırma başarısının tam doğru olduğunu göstermektedir. MCC değeri şu şekilde hesaplanır [17]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (3)$$

+1 değerine yakın MCC sonucu sınıflandırıcının oldukça başarılı bir sonuç verdiğini gösterir.

4. DENEYSEL SONUÇLAR(EXPERIMENTAL RESULTS)

4.1.Veri Kümelerinin Hazırlanması (Dataset Preperation)

UCI veri setleri ortak kullanıma açıktır ve gerçek hayattan alınan verilerle oluşturulmuştur. Bu veri setlerindeki sınıf türleri genel olarak dengeli bir dağılıma sahiptir. Ancak 10 adet iki sınıflı gerçek veri seti eşit bir dağılıma sahip olmadığı için seçilerek çalışmamıza dâhil edilmiştir. Çok sınıflı veri setlerinden bazıları seçilerek bu veri setlerinde bulunan iki sınıf türüne ait örnekler alınıp yeni bir veri setine aktarılmıştır. Bu sayede 8 adet dengesiz veri seti yapay olarak türetilmiştir. Veri setlerinde bulunan tüm nominal değerler ikili sayısal değerlere dönüştürülmüş kayıp değerler yer değiştirilmiş, ve normalizasyon işlemi yapılmıştır. Bu işlemdeki amaç MCC performans ölçütünün dengesiz veriler üzerinde kullanılabilmesini sağlamaktır. Tablo 1'de oluşturulan veri setlerinin isimleri, örnek sayıları, her bir sınıfa ait örnek dağılımları ve Zero Rule kuralına göre rastgele sınıflandırma başarıları verilmiştir. Veri setleri *arff* formatındadır.

Tablo 2. Kullanılan veri setleri
(Datasets used in the experiments)

Name	N	C_1	C_2	ZeroRule Acc
anneal_a	783	99	684	0.87
anneal_b	724	40	684	0.94
anneal_c	751	67	684	0.91
autos_a	89	22	67	0.75
balance-scale_a	337	49	288	0.85
balance-scale_b	337	49	288	0.85
breast-cancer	286	85	201	0.70
breast-w	699	241	458	0.66
colic	368	136	232	0.63
credit-g	1000	300	700	0.70
diabetes	768	268	500	0.65
glass_a	99	29	70	0.71
hepatitis	155	32	123	0.79
Hypothyroid_a	3675	194	3481	0.95
ionosphere	351	126	225	0.64
labor	57	20	37	0.65
sick	3772	231	3541	0.94
zoo_a	51	10	41	0.80

4.2.Algoritmaların Uygulanması (Applications of Algorithms)

Weka yazılımı ve MATLAB programlama dili ile yukarıda bahsi geçen sınıflandırıcıların, 18 adet veri seti üzerindeki performans analizleri yapılmıştır. Belirtildiği üzere tüm algoritmaların başarı analizleri 10 kat çapraz geçerleme ve MCC kriteri ile yapılmıştır.

Tablo 3. Algoritmaların parametreleri
(Parameters of Algorithms)

	Temel Öğrenici	İtererasyon sayısı (I)	Veriseti Sayısı
DT	CART	1	1
Bagging	CART	-	20
Boosting	CART	20	1
Random Subspaces	Linear Discriminant	20	1
Dagging	J48	-	20
Decorate	J48	-	20

Tablo 3'de her bir algoritmanın hangi parametreleri kullandığı belirtilmiştir. Rondom Subspace yöntemi hariç diğer tüm yöntemler temel öğrenici olarak karar ağaçlarını kullanmıştır. Bilindiği üzere J48, Weka'da karar ağaçlarının adıdır. Tüm kolektif öğrenicilerin birbiri ile karşılaştırılabilmesi için iterasyon sayıları da eşit alınmıştır.

Tablo 4. Kolektif sınıflandırıcıların MCC değerleri
(MCC results of the ensemble methods)

	Karar Ağacı	Bagging	AdaBoost	Dagging	Decorate	Random Subspace
anneal_a	**0.99**	0.97	0.98	0.96	0.97	0.96
anneal_b	**1.00**	0.98	0.98	**1.00**	0.99	1.**00**
anneal_c	0.93	0.85	0.96	0.90	0.90	0.95
autos_a	0.79	0.69	**0.88**	0.79	0.72	0.81
balance-scale_a	0.14	0.07	**0.42**	0.10	0.05	0.02
balance-scale_b	0.29	0.07	**0.38**	0.14	0.08	0.12
breast-cancer	0.17	0.18	**0.29**	0.24	0.22	0.24
breast-w	0.83	0.90	0.90	0.90	0.90	**0.91**
colic	0.61	0.63	**0.65**	0.64	0.60	0.64
credit-g	0.26	**0.38**	**0.38**	0.28	0.32	0.26
diabetes	0.34	**0.47**	0.56	0.45	0.42	0.46
glass_a	0.88	0.93	**0.95**	0.93	0.90	0.93
hepatitis	0.38	**0.53**	0.52	0.41	0.41	0.43
hypothyroid_a	**0.99**	0.43	0.98	0.97	0.97	0.97
ionosphere	0.75	0.71	0.84	0.85	0.78	**0.86**
labor	0.69	0.60	0.70	**0.71**	0.69	0.69
sick	**0.87**	0.58	0.79	0.63	0.75	0.81
zoo_a	**1.00**	**1.00**	0.98	**1.00**	**1.00**	**1.00**
ORTALAMA	0.66	0.62	0.74	0.63	0.64	0.67

978-1-4673-7386-9/15/$31.00 ©2016 IEEE

4.3.Deneysel Sonuçlar (Experimental Results)

Tablo 4'de biri tekil öğrenici olan karar ağacı tekniği ile 5 adet farklı kolektif sınıflandırıcının veri setleri üzerindeki başarılarına yer verilmiştir.

Tablo 4'nin her bir satırda bulunan veri seti için en yüksek sınıflandırma başarısı gösteren kolektif yöntemi vurgulamak için başarı oranı koyu renk ile gösterilmiştir. Buna göre AdaBoost yöntemi diğerlerine göre en fazla yüksek başarıyı elde eden algoritma olmuştur. En son satırda ise her bir algoritmanın başarıları ortalama olarak verilmiş ve bu sayede hangi algoritmanın genel olarak daha başarılı olduğu anlaşılmaya çalışılmıştır.

4.4.FRIEDMAN Testi (FRIEDMAN Test)

Her bir öğrenici grubunun ayrı ayrı veri kümeleri üzerinde verdiği bu sonuçları bir birleri ile kıyaslayarak en başarılı kolektif öğrenicinin tespit edilmesi istenmiştir. Bu amaçla dengesiz veri kümeleri üzerinde en başarılı yöntemi tespit etmek için FRIEDMAN testi kullanılmıştır. Literatürde FRIEDMAN testi ANOVA'ya göre daha güvenilir sonuçlar verdiği için tavsiye edilmektedir[18].İstatistiksel bir yöntem olan bu testte birçok sınıflandırıcının elde ettiği başarı değerlerinin karşılaştırması yapılmaktadır. Bu amaçla FRIEDMAN testi için hazırlanmış MATLAB kodları kullanılmıştır [19].

Her bir algoritmanın diğer algoritmalar üzerinde istatistiksel karşılaştırmalar neticesinde sağladıkları üstünlük sayısı aşağıdaki tabloda verilmiştir. A1, A2, A3, A4, A5 ve A6 algoritmaları sırasıyla DT, Bagging, Boosting, Random Subspaces, Dagging ve Decorate yöntemleridir. Bu tabloda satırlar hep bir algoritmanın diğerlerine göre sağladıkları üstünlük sayılarını; sütunlar ise her bir algoritmanın diğerlerin karşısında kayıp sayılarını göstermektedir.

Tablo 5. Algoritmaların üstünlük sayıları
(Win numbers of each algorithm)

	A1	A2	A3	A4	A5	A6
A1	-	3	0	0	2	1
A2	1	-	0	1	2	1
A3	7	14	-	9	7	5
A4	3	1	1	-	0	1
A5	3	2	0	0	-	2
A6	2	1	3	2	1	-

A3 yöntemi yani AdaBoost algoritması yatay olarak incelendiğinde A1, A2, A4, A5 ve A6'ya göre sırasıyla 7, 14, 9, ve 5 defa üstün gelmiştir. A3 sütunu incelendiğinde ise A4 ve A6'ya toplamda 4 defa yenilmiştir. Burada iki algoritma birbiri ile karşılaştırılırken belirli bir veri seti üzerinde elde edilen 10 Kat çapraz geçerleme sonucu ele alınarak yapılmıştır. Her bir algoritma için yapılan 10 Kat çapraz geçerleme işlemi sonucunda elde edilen 10 adet yüzdelik başarı değeri diğer teknikler ile karşılaştırılırken istatistiksel anlamlılığın belirlenmesinde kullanılmaktadır.

FRIEDMAN testindeki algoritmaların ortalama seviye (*Average ranks*) değerleri ise Tablo &'da verilmiştir.

Tablo 6. Algoritmaların FRIEDMAN testi sonuçları
(FRIEDMAN Test Results of Algorithms)

	A1	A2	A3	A4	A5	A6
Test Sonuçları	0,92	0,76	2,21	0,81	0,71	1,08

Tablo 6'da da görüldüğü üzere elde edilen bu sonuçlara göre dengesiz veri setleri üzerinde sınıflandırıcı topluluklarının başarı sıralaması yapılacak olursa sırasıyla AdaBoost, Decorate, Karar Ağaçları, Rastsal Alt uzaylar, Baggin ve Dagging'dir. Diğer algoritmalarla kıyaslandığında 2,21'lik test sonucu ile AdaBoost yönteminin en yüksek performansı gösteren sınıflandırıcı olduğu görülmektedir. AdaBosst ve Decorate dışında diğer kolektif yöntemler tekil bir sınıflandırıcı olan karar ağaçlarından daha düşük bir performansa sahiptir. Ayrıca Baggin ve Dagging yöntemleri çalışma prensipleri itibarıyla birbirlerine benzedikleri için benzer bir başarı oranına sahiptir.

5. DEGERLENDİRME (EVALUATIONS)

Temel istatistiksel değerler ve FRIEDMAN testi sonuçları baz alındığında AdaBoost yönteminin, temel öğrenici olan karar ağacı ve diğer öğrenici topluluklarına göre en başarılı yöntem olduğu anlaşılmıştır. Bunun nedenini şu şekilde açıklamak mümkün olabilir. AdaBoost, her bir çevrimde kendini seyrek dağılıma sahip sınıf türüne karşı eğitebildiği için toplamda daha yüksek başarı sunabilmiştir.

Bilindiği üzere Bagging, Dagging ve Decorate yöntemlerinde yapay yeni veri setleri türeterek aynı sınıflandırıcının eğitilmesi söz konusudur. Bu üç yöntem görüldüğü üzere birbirlerine yakın başarı göstermişlerdir. Sınıf dağılımı dengeli olmayan bir veri setinden değişik yöntemlerle yeni veri setleri türetildiğinde de dengesiz yeni veri kümeleri ortaya çıkmaktadır. Dağılımları dengesiz veri setlerinde de yeteri düzeyde farklılıklar (*diversity*) oluşturulamayacağı için toplam başarı yükselemeyecek aksine düşecektir. Bu yöntemlerde temel öğreniciler kendilerini seyrek sınıf türüne göre yeteri düzeyde eğitememekte ve istenilen kümülatif başarıya ulaşılamamaktadır. Bu sebeplerden ötürü Bagging, Dagging ve Decorate düşük performans sergilemiştir. AdaBoost yöntemi bu üç algoritmadan farklı olarak, veri setinden rastgele seçim yaparak yeni sınıf türleri oluşturmak yerine hatalı sınıflandırma işleminin hangi

örnekler ile yapıldığını tespit etmekte ve seçimleri bu örmekler üzerinden yapmaktadır. Bu sayede AdaBoost diğerlerine göre daha yüksek bir başarı sunmuştur.

Random Subspace yönteminde bir veri setine ait özniteliklerden bazıları silinerek yeni veri setleri türetilmektedir. Bu işlemde oluşan daha az boyutlu yeni veri setlerinde belirli bir sınıfa ait örnek sayıları azınlıkta kalmaya devam edecektir. Daha önce olduğu gibi öğrenici kendini çoğunlukta olan sınıfa göre eğitecek ve sistemde ister istemez önyargı durumu kaybolmayacaktır.

Çalışmamızda değişik durumlarda karşımıza çıkabilecek ve gerçek hayatta var olan dengesiz veri setlerine karşı yüksek doğruluk oranı veren sınıflandırıcı gruplarının deneysel analizi yapılmış ve başarı sıralaması gerçekleştirilmiştir. Literatürde önerilmiş değişik sınıflandırıcı topluluklarıyla sınıf dağılımları dengesiz olan veri setleri üzerinde uygun istatistiksel ölçütlerle performans analizi yapılmıştır. Karşılaştırılan yöntemler içerisinde en yüksek performansı sergileyen algoritma AdaBost olmuştur. Sınıflandırıcı topluluklarının dengesiz veri setleri üzerindeki başarı ve başarısızlıkları üzerine bir takım değerlendirmeler de yapılmıştır.

KAYNAKLAR (REFERENCES)

[1] Wu, G., Chang, E.Y., "**Class-boundary alignment for imbalanced dataset learning"**. In ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC, 49-56, 2003.

[2] He H., "**Learning from Imbalanced Data**", IEEE Transactions on Knowledge and Data Engineering, 21(9):1263 – 1284, 2009.

[3] Han H., Wang W.Y., Mao B.H., **Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning**, Springer Berlin Heidelberg 5, 3644: 878-887, 2014.

[4] Akbani R., Kwek S., Japkowicz N., "**Applying Support Vector Machines to Imbalanced Datasets**", Machine Learning: ECML of the series Lecture Notes in Computer Science, vol 3201:39-50, 2004.

[5] Kotsiantis S., Kanellopoulos D., Pintelas P., "**Handling imbalanced datasets: A review**" , GESTS International Transactions on Computer Science and Engineering, Vol.30, 2006.

[6] Jian, C., Gao, J., Ao, Y., "**A new sampling method for classifying imbalanced data based on support vector machine ensemble"**, Elsevier Neurocomputing Journal, 1-8, 2016.

[7] Haixiang, G., Yijing, L., Yanan, L., Xiao, L., Jinling, L., "**BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification"**, Elsevier Engineering Applications of Artificial Intelligence, Vol 49, 176-193, 2016.

[8] Yijing, L., Haixiang, G., Xiao, L., Yanan, L., & Jinling, L., "**Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data"**, Elsevier Knowledge-Based Systems, 94, 88-104, 2016.

[9] Zhou, Zhi-Hua. "**Ensemble methods: foundations and algorithms**", CRC Press, New York, ABD, 2012.

[10] Breiman L., "**Bagging predictors**.", Machine Learning, 24:123–140, 1999.

[11] Freund, Y., "**A short introduction to boosting**", Journal-Japanese Society For Artificial Intelligence, 14(1):771-780, 1999.

[12] Breiman, L., "**Bias, Variance, and Arcing Classifiers**", Technical Report, STATISTICS Department, University Of California, Berkeley, 1996.

[13] Tetik, Y.E., "**Gürültülü Ortamlarda Konuşma Tespiti İçin Yenibir Öznitelik Çıkarım Yöntemi**". Elektrik-Elektronik ve Bilgisayar Sempozyumu. Fırat Üniversitesi-Elazığ, 2011.

[14] Ting, K. M., Witten, I. H.: "**Stacking Bagged and Dagged Models**". 14th International Conference on Machine Learning, San Francisco, CA, 367-375, 1997.

[15] Melville P., Mooney R.J.,"**Constructing Diverse Classifier Ensembles Using Artificial Training Examples**". Eighteenth International Joint Conference on Artificial Intelligence, 505-510, 2003.

[16] Ho T.K., "**The Random Subspace Method for Constructing Decision**", IEEE Transactions on Pattern Analysis and Machine Intelligence, Lucent Tech no l., AT&T Bell Labs., Murray Hill, 20(8):832 – 844, 1998.

[17] Matthews B.W., "**Comparison of the predicted and observed secondary structure of T4 phage lysozyme**". Biochimica et Biophysica Acta (BBA) 405 (2): 442–451, 1975.

[18] Demsar J., "**Statistical Comparisons of Classifiers over Multiple Data Sets**", Journal of Machine Learning Research 7:1–30, 2006.

[19] Ulaş A., Yıldız O.T., Alpaydın E., "**Cost-conscious comparison of supervised learning algorithms over multiple data sets"**, Pattern Recognition, 45(4): 1772–1781, 2012.