Makale Türü: Araştırma Makalesi
Gönderim Tarihi: 25.09.2020 **Kabul Tarihi:** 30.12.2020

Sınıflandırma Başarımını Ölçme ve Seyreklik İşleme Üzerine
On Evaluating Classification Performance and Handling Rarity

Umut Konur

Zonguldak Bülent Ecevit Üniversitesi Bilgisayar Mühendisliği Bölümü Zonguldak, Türkiye
konur@beun.edu.tr

Öz

Sezgisel olarak, sınıflandırıcı başarımını ölçme, sınama örnekleri üzerinde koşma yaparak ve doğru kararların tüm kararlara oranı gözlemlenerek yapılabilir görünmektedir. Ne var ki, çoğu zaman eldeki probleme bağlı olan, doğru ve yanlış kararların bağıl önemi (ağırlığı) ve çeşitli işletim noktaları dikkate alındığında, konu bu kadar basit değildir. Genellikle belli bir ölçev türünden yüksek başarım istenirken, başka bir ilintili ölçev türünden daha düşük bir başarım kabul edilebilmektedir. Böyle bakılırsa, başarım ölçme, mühendislik görüngülerinin çoğundaki gibi, bir ödünleşim süreci olarak anlaşılabilir. Bu makalede, sınıflandırıcıların karşı karşıya olduğu farklı koşullar da dikkate alınarak başarım ölçme üzerine bir derleme sunulurken, sınıflandırma başarımını azaltan ve benzer özellikler barındıran durum/sınıf seyrekliği ve sınıf dengesizliği (dengelenmemiş durum/sınıf dağılımları) sorunları üzerinde durulmaktadır. Derleme, bu bağlamda, söz konusu darboğazların üstesinden gelmeyi amaçlayan tamamlayıcı bir örnek yöntemler kümesi sunmaktadır.

Anahtar kelimeler: ölçev, alıcı işletim eğrisi, seyreklik, sınıf dengesizliği, tümevarımsal yanlılık, örnekleme, iteleme, öznitelik seçimi, karma öğrenme, alan bilgisi

Abstract

Evaluating classifier performance intuitively seems to be achievable by running it on test samples and observing the fraction of correct decisions. However, this matter is not that simple when one considers the relative importance (weight) of correct and incorrect decisions, usually depending on the problem at hand, and various operating points. One can generally desire high performance in terms of a specific metric whereas a lower performance with another related metric can be accepted. From this point of view, performance evaluation can be conceived as a trade-off process, as in most engineering phenomena. In this article, while a review of performance evaluation metrics considering different conditions that classifiers are subject to is presented, the similar problems of case/class rarity and class imbalance (imbalanced case/class distributions) which degrade classifier performance are dealt with. Within this context, the review presents a complementary set of representative methods to tackle these bottlenecks.

Keywords: metric, receiver operating characteristics, rarity, class imbalance, inductive bias, sampling, boosting, feature selection, hybrid learning, domain knowledge

1. Giriş

Sınıflandırıcılar hayatımızın vazgeçilmezleri arasıa girmiştir. Uydu imgelerinden petrol saçılımların belirleme [1], belge kategorizasyonu [2], hava tahminleri [3], ürünlerdeki kusurları bulma [4], bilgisayar ağ saldırılarını ayırt etme [5], müşteri risk analizi [6], hastalık sezimi yapma [7] gibi birçok alanda; genellikle tanısal sistemler biçiminde karşımıza çıkan sınıflandırıcıların başarımını nicel olarak ortaya koyabilmek, ilk etapta düşünüldüğünün aksine, kolay bir problem olarak ele alınmamalıdır. Sınıflandırıcıların tasarımı aşamasında kullanılan öğrenme kümelerinde, var olan sınıfların tümüne ait yeterli örneğin olmaması ve bu yüzden öznitelik uzayının da problem açısından dengeli biçimde taranamayıp göstericiliğin azalması zorluğu belirginleştirmektedir. Başarım yönünden, beklentiler farklı olabilir. Örneğin, bir kategoriye ait örnekleri yanlış sınıflandırmak katlanılamaz iken, başka kategoriler için durum bu kadar önemli olmayabilir. Sınıflandırıcıların başarımını görmek, koşma koşullarını düzenlemek ve koşullara bağlı olarak farklı seçimler arasından en iyi sınıflandırıcıyı tercih etmek için en az bir araç bulunması da sınıflandırıcı tasarımında önemlidir.

Bu çalışmada, başarım ölçevleriyle ilgili kısa açıklamalar yapılmakta, başarımı farklı işletim noktalarında değerlendiren alıcı işletim eğrisi (*ROC*) [8] çözümlemesi üzerinde durulmakta, öğrenme aşamasında seyrek (dengesiz dağılımlı) veri kullanan sınıflandırıcıların bu problemle nasıl baş edebilecekleri konusunda bir derleme sunulmakta ve genel ilkeler olarak işe yaraması beklenen sonuçlara varılmaktadır.

2. Başarım Ölçevleri

Sınıflandırma başarımını ölçme, en temelde, söz konusu sınıflandırıcının sınama örneklerinin hangi sınıfa ait olduklarına dair kararlarını örneklere ait doğru sınıf etiketleriyle karşılaştırmak yoluyla yapılmaktadır. Başarım bildiriminde yararlanılan her ölçev, bu karşılaştırmaların sonuçlarının fonksiyonları olarak formule edilmektedir. Basitlik açısından sınıflandırma problemleri iki sınıfa indirgenip tanısal problemler olarak ifade edildiğinde, iki ayrı sınıftan söz edilir: *pozitif* ve *negatif*. Sezimi daha önemli olan sınıf pozitif sınıf, diğeri negatif sınıf olarak adlandırılır. Bir örneğe dair sınıflandırıcının pozitif veya negatif olarak karar verdiği etiket örneğin etiketiyle aynıysa, sınıflandırıcının örnek üzerinde başarılı, aksi durumda başarısız olduğu söylenir. Her örnek için iki olası olay gerçekleşmesi ve iki olası karar bulunan tanısal bir sistemde, sınıflandırma herbiri

başarı ya da başarısızlık olarak değerlendirilen dört farklı sonuç verebilir. *TP* (true positive) pozitif olarak karar verilen pozitif örnekleri, *FP* (false positive) pozitif olarak karar verilen negatif örnekleri, *FN* (false negative) negatif olarak karar verilen pozitif örnekleri ve *TN* (true negative) negatif olarak karar verilen negatif örnekleri göstermektedir. Bu durumların gerçekleşme sayıları olan *TPs*, *FPs*, *FNs* ve *TNs* değerleri bir olasılık tablosu (olasılık matrisi) [9] olarak gösterilebilir. Tablo 1 iki-sınıflı bir probleme ait olasılık tablosudur. Sütunlar gerçek olaylarla (örneklerin pozitif veya negatif olmasıyla), satırlar ise örneklere ait kararlarla ilgilidir.

Tablo 1: Olasılık tablosu

		Olay	
		pozitif	negatif
Karar	pozitif	*TPs*	*FPs*
	negatif	*FNs*	*TNs*

Os örnek sayısını, *Ps* pozitif örneklerin sayısını, *Ns* de negatif örneklerin sayısını gösterirken aşağıdakiler geçerlidir:

$$Ps = TPs + FNs \tag{1}$$

$$Ns = TNs + FPs \tag{2}$$

$$Os = Ps + Ns \tag{3}$$

Yanlış negatiflere (*FN*), sıklıkla, Tip I hata; yanlış pozitiflere (*FP*) ise Tip II hata denir.

Veri madenciliği [10], yapay öğrenme [11] gibi alanlarda kullanılan en temel ölçev olan *doğruluk* (accuracy) *A*, doğru kararlarının sayısının tüm örneklerin sayısına oranıdır:

$$A = \frac{TPs + TNs}{TPs + TNs + FPs + FNs} \tag{4}$$

İşe yarar görünmekle beraber, dengelenmemiş dağılımlı veri kümelerinde, *A* değerlerinin büyüklüğü daha iyi başarım anlamına gelmeyebilir. Örneğin; 99 negatif, 1 pozitif örnekten oluşan veri için bütün örnekleri negatif olarak sınıflayan bir sistemde, %99 başarım görülecek ancak seyrek olan pozitif örneğe ait karar yanlış olduğu için, sistem pratikte başarısız olarak değerlendirilecektir. Kaldı ki, böyle problemlerde değerli olan seyrek sınıfa ait örneklere ait kararların doğru verilebilmesi ve sıradışı (seyrek) örneklerin sezilebilmesidir.

Duyarlık (sensitivity) olarak da bilinen *geri getirme* (recall) *R* ve *kesinlik* (precision) *P* başlı başına kendi değerleriyle veya diğer ölçevlerin hesaplanmasında kullanılabilecek temel başarım ölçevlerindendir. Sınıflandırma başarımını ölçmede amaç, başarım bildirimini çoğunlukta olan (örnek sayısı diğer sınıfınkilerden oldukça fazla olan) sınıfa göre yapmak değildir. Aksine, seyrek sınıf üzerindeki yüksek başarım daha değerlidir. Pozitif sınıf üyelerine göre tanımlanan geri getirme R_p ve kesinlik P_p, sırasıyla (5) ve (6)'da verilmektedir:

$$R_p = \frac{TPs}{Ps} = \frac{TPs}{TPs + FNs} \tag{5}$$

$$P_p = \frac{TPs}{TPs + FPs} \tag{6}$$

Geri getirme ve kesinlik ölçevleri negatif sınıf üyelerine göre tanımlandığında (R_n ve P_n) gösterim (7) ve (8)'deki gibidir:

$$R_n = \frac{TNs}{Ns} = \frac{TNs}{TNs + FPs} \tag{7}$$

$$P_n = \frac{TNs}{TNs + FNs} \tag{8}$$

Pozitif örnekler için geri getirme R_p, doğru sınıflandırılan pozitif örneklerin sayısının bütün pozitif örneklerin sayısına oranına, kesinlik P_p de doğru yapılan pozitif sınıflandırmaların sayısının tüm pozitif sınıflandırmaların sayısına oranına karşılık gelir. Doğru sınıflandırılan negatif örneklerin sayısının negatiflerin toplam sayısına oranı (negatifler için geri getirme R_n) *özgüllük* (specificity) olarak adlandırılır. Pozitifler için geri getirme, negatiflerin ne kadarının yanlış sınıflandırıldığı konusunda bir görüş sağlamazken, kesinlik pozitiflerin ne kadarının negatif olarak sınıflandırıldığının göstergesi değildir. Geri getirme ve kesinlik ölçevlerini beraber kullanmak, dengesiz veriyle öğrenen sınıflandırıcılar için, başarımı ortaya koymanın etkin bir yoludur.

Bilgiye geri erişim [12] alanında sıklıkla kullanılan *F-ölçevi* (F-measure) *F*, geri getirme ve kesinliğin bağıl önemlerini ağırlıklandırır. *R* iki-sınıflı bir problemde herhangi bir sınıf için geri getirmeyi, *P* de kesinliği gösterirken, bu sınıf için *F*'nin genel biçimi aşağıdadır:

$$F = \frac{1}{\lambda \frac{1}{R} + (1-\lambda)\frac{1}{P}}, 0 \le \lambda \le 1 \tag{9}$$

Pozitif sınıf üyeleri için F-ölçevi F_p ele alındığında; λ değerini artırmak, geri getirmeye verilen önemi artırmakta (yanlış negatifler *FN*'yi azaltmak amacıyla) ve λ değerini azaltmak kesinliğe verilen önemi artırmaktadır (yanlış pozitifler *FP*'yi azaltmak amacıyla). Geri getirme ve kesinliğe eşit önem verildiğinde (λ= 0.5), (9) aşağıdaki gibi yazılabilir:

$$F = \frac{2RP}{R + P} \tag{10}$$

Bir diğer yaygın başarım ölçevi (11)'deki herhangi bir sınıf için *geri getirme ve kesinliğin geometrik ortası* (Geometric Mean of Recall and Precision: *GMRP*)dır. Benzer biçimde, iki sınıflı problemlerde, *geri getirmelerin geometrik ortası* (Geometric Mean Of Recalls: *GMOR*), (12)'deki gibidir:

$$GMRP = \sqrt{RP} \tag{11}$$

$$GMOR = \sqrt{R_p R_n} \tag{12}$$

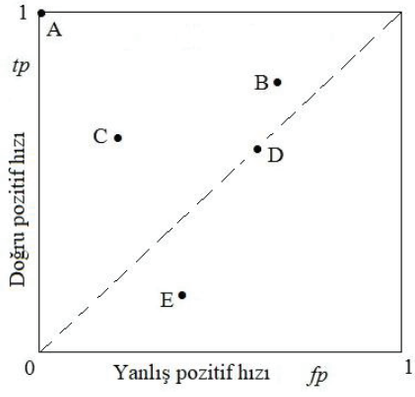
Geri getirmelerin toplamı (sum of recalls) ölçevi *SOR*, pozitif ve negatif sınıflar için geri getirmelerin toplamıdır. Bu ölçeve *ağırlıklı doğruluk* da denir:

$$SOR = R_p + R_n \tag{13}$$

Sözü edilen ölçevlerin yanı sıra, özellikle seyrek sınıf problemlerindeki sınıflandırma başarımını bilgi-kuramsal ölçen bilgi puanı [13], seyrek sınıfın doğru sınıflandırılmasını çoğunluktaki sınıfın doğrı sınıflandırılmasından daha fazla ödüllendiren, benzer biçimde çoğunluktaki sınıfın yanlış sınıflandırılmasını seyrek sınıfın yanlış sınıflandırılmasından daha çok cezalandıran, bu bağlamda iki sınıfı da dikkate aldığı için bütüncül başarımı tek bir nicelik olarak ortaya koyan bir ölçevdir. Geri getirme ve kesinlik ölçevlerinin bir ödünleşim ilişkisi içinde olduğu gözlemine dayanan, bu iki ölçevin eşitlenebileceği düzeyi belirleyen *eşitleme noktası* (break-even point) kimi zaman kullanılan bir diğer ölçevdir.

3. ROC Çözümlemesi

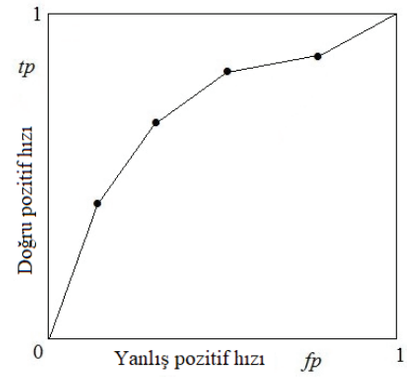
Önceki bölümdeki ölçevler, sınıflandırma başarımını tek bir işleyiş noktasında tanımlamak için kullanılır. Buna karşın, sınıflandırıcıların başarım değerlendirmesi, herbiri yanlış pozitif kararlarının çoğalması pahasına başarılan daha fazla doğru pozitif kararları ile belirlenen farklı noktalar (bir işleyiş noktaları eğrisi) boyunca yapılabilir. Sınıflandırıcı başarımının değerlendirilmesi için *alıcı işletim eğrisi* (receiver operating characteristics: *ROC*) [8] adı verilen kullanışlı bir araç bulunmaktadır. Söz konusu eğri üstündeki herbir nokta bir doğru pozitif hızını (*tp*) ve karşılık gelen bir yanlış pozitif hızını (*fp*) göstermektedir. Her noktanın ayrı bir işletim noktası olduğu eğri boyunca yapılan inceleme (ROC çözümlemesi) sınıflandırıcıların tasarımı ve parametrelerin belirlenmesi sürecinde işe yaramaktadır. İki-boyutlu (2D) çizgelerle görselleştirilen ROC uzayında, *fp* ve *tp* sırasıyla *x* ve *y* eksenlerine karşılık gelir. Şekil 1, beş ayrı sınıflandırıcının (A, B, C, D, E) işaretlendiği bir ROC çizgesidir.



Şekil 1: A, B, C, D, E sınıflandırıcılarının ROC çizgesi

Kusursuz bir sınıflandırıcı, hiç yanlış pozitif kararı vermeden bütün pozitifleri doğru sınıflandırandır. Şekil 1'deki A, $fp = 0$ ve $tp = 1$ değerlerine sahip (*ROC* çizgesindeki (0, 1) noktası) kusursuz bir sınıflandırıcıdır. Sınıflandırıcıları karşılaştırıp "daha iyi" olanlarını seçerken, yüksek *tp*'ye ve düşük *fp*'ye sahip olmaları istenir. Bu gözlem, bir sınıflandırıcıya ait *ROC* noktası başka birisine ait *ROC* noktasının kuzey-batısında ise, ilkinin başarımının ikincininkinden daha iyi olduğu anlamına gelir. Şekil 1'deki A'nın bütün sınıflandırıcıların en iyisi olduğu ve C'nin D'den daha iyi olduğu sonuçlarına varılabilir. (0, 0) noktası, o noktada işleyen bir sınıflandırıcının hiç doğru pozitif karar vermediğini ve hiç yanlış pozitif karara yol açmadığını göstermektedir. Aslında, (0, 0) herhangi bir sınıflandırıcının hiç karar vermediği bir noktadır (var olmayan bir sınıflandırıcı). Diğer uç nokta (1, 1), o noktada işleyen bir sınıflandırıcının bütün örnekler için pozitif olma kararı verdiği, bu yüzden de bütün pozitifleri doğru sınıflandırırken bütün negatifleri de yanlış sınıflandırdığı anlamına gelir. Pozitif kararlarını kolay vermek ((1, 1) noktasındaki gibi) ve zor vermek (yalnız güçlü delil varsa), sınıflandırıcıların "hür (liberal)" veya "muhafazakar (conservative)" olarak nitelendirilebileceğini göstermektedir [8]. Hür bir sınıflandırıcı, doğru pozitif kararlarını daha fazla vermekle birlikte daha fazla yanlış pozitif hataları yapar. Bunun tersine, muhafazakar sınıflandırıcılar örneklerin pozitif olduğuna sık olarak karar vermez ve yanlış pozitif hataları da doğru pozitif kararları da az olur. Şekil 1'deki C, B'den muhafazakardır (veya, B, C'den hürdür). Sınıflandırıcıların köşegen üstündekileri ($y = x$), rasgele karar verenler olarak görülebilir. Kararlarının yarısı pozitif diğer yarısı negatif olan bir sınıflandırıcı (0.5, 0.5) noktasındadır. Benzer biçimde, kararlarının %20'si pozitif olan bir sınıflandırıcı (0.2, 0.2) noktasında işlemektedir. D, rasgele karar veren bir sınıflandırıcıdır. Bir sınıflandırıcı (Şekil 1'deki E gibi), ROC köşegeninin altında ise, gerçekçi değildir, çünkü başarımı rasgele karar verenlerinkinden azdır ve kararların tersini almak köşegenin yukarısında yer alan bir sınıflandırıcı üretecektir.

Çoğu sınıflandırıcı, ROC uzayında tek nokta verecek şekilde tasarlanmaktadır. Diğer yandan, bazı sınıflandırıcılar bir tür puan üretir (örnek olasılığı gibi) ve kararlarını bu puana dayandırır. Puanı kullanan eşik değerleri sınıflandırma kuralını değiştirmek üzere ayarlandığında, ROC çizgesinde, puanı farklı eşik değerleriyle kullanan bir sınıflandırıcıya (farklı işletim noktalarında çalıştırılan bir sınıflandırıcıya) ait farklı noktalar elde edilir. Bir sınıflandırıcıya ait ROC çizgesindeki farklı işletim noktalarına karşılık gelen noktalar birleştirilince, *ROC* uzayında bağlı bir eğri oluşmakta ve böyle eğriler *fp* ve *tp* değerleri boyunca sınıflandırma başarımına gösterge olmaktadır. Bir ROC eğrisi oluşturma, puanı uygun değerlerde örneklemek, sınıflandırıcıyı bütün örneklenen değerler için (bütün işletim noktalarında) koşmak ve elde edilen noktaları uygun biçimde bağlamakla başarılır. ROC noktalarını ((*fp*, *tp*) ikilileri) bağlamak için doğrusal aradeğerleme kullanılırsa, bu noktalar, en küçük *fp* değerine sahip noktadan başlayarak en büyük *fp* değerine sahip noktaya doğru sırayla bağlanır. *x* ve *y* eksenlerinin bütün erimini kapsamak için, en soldaki noktayı (0, 0) noktasına, en sağdakini de (1, 1) noktasına bağlamak gerekir. Şekil 2, dört noktada işletilen bir sınıflandırıcının ROC çizgesidir.



Şekil 2: Dört noktada işleyen bir sınıflandırıcının ROC çizgesi

ROC eğrileri kullanılarak, eğri ve *x* ekseni (*fp*) arasındaki alanın değerine sahip *ROC eğrisi altındaki alan* (area under the ROC curve: *AUC*) [14] başarım ölçevi tanımlanır. *AUC*, tek bir noktayı değil bütün eğriyi gözetir ve eğri boyunca sınıflandırıcının bütüncül başarımını ölçer. Daha büyük *AUC* değerleri daha yüksek başarımı göstermektedir. Gerçekçi bir sınıflandırıcının *AUC* değeri 0.5'ten küçük olamaz.

4. Seyreklik Madenciliği

Veri madenciliği problemlerindeki seyreklik sorunu, temelde *seyrek sınıflar* (yaygın olarak *sınıf dengesizliği*) ve *seyrek durumlar* olmak üzere ikiye ayrılmaktadır [15]. Seyrek sınıflar, etiketlenmiş örnekler bulunan sınıflandırma ile ilgilidir. Fetal kafataslarının ultrason imgelerinden spina bifida patolojisini sezme probleminde [16][17], 358 kafatası imgesinden 329'u sağlıklı ve 29'u hastalıklı etiketlenmiştir. 358 imgeden yalnızca 29'unda bulunan "hastalıklı kafatasları" iyi bir seyrek sınıf örneğidir. Buna karşın, seyrek durumlar verinin anlamlı ama görece küçük bir altkümesine (örnek uzayının küçük bir bölgesine) karşılık gelir. Seyrek durumlar, yalnızca veri dağılımına bağlı oldukları için hem etiketlenmiş hem de etiketlenmemiş veri için tanımlanabilir. Örneğin, etiketlenmiş veriyle ilgilenilen durumda, seyrek bir durum az bulunan bir altsınıfı gösterebilir. Seyrek durumları belirlemek çoğu zaman zor olup gruplandırma (clustering) gibi öğreticisiz teknikler bu durumların saptanmasında yararlı olabilir. Bir sınıflandırıcının, az sayıda öğrenme örneği içeren küçük bölgeleri de seyrek durumları ortaya koyabilir. Seyrek sınıfların ve durumların tanımları arasında kavramsal bir fark bulunmakla beraber, her ikisi için de karşılaşılan problemler ve çözüm yaklaşımları benzerlik gösterir.

Bir düzey daha ileri gidilirse, seyrek sınıflar/durumlar iki farklı çerçevede anlaşılabilir. Sınıf/durum örneklerinin sayısının *mutlak* anlamda küçük olması (veriden yoksun olma) ve bazı sınıf/durum örneklerinin diğerlerine göre azlığı anlamına gelen *görece* seyreklik (sınıf/durum dengesizliği), verideki düzgünlükleri saptama açısından sorunlara yol açmakta, böylelikle yetersiz veya yanıltıcı öğrenmeye ve yanlış genelleştirmeye neden olmaktadır.

Mutlak veya görece seyreklik içeren veri madenciliğinde zorlaştırıcı etkenler incelendiğinde, yöntemleri yönlendirmede ve sonuçları değerlendirmede kullanılan ölçevlerin seçiminin (seçimler aynı yöntemin farklı aşamalarında farklı olabilir) ciddi önemi olduğu görülür. Ayrıca, seyrekliğe böl-ve-yönet yöntemleriyle çözüm arayan öğrenme algoritmalarının *veri parçalanması*na ve düzgünlüklerin yalnızca daha az veri içeren bölüntülerde bulunmasına yol açtığı bilinir. Sonuç olarak, seyreklik içeren problemler böl-ve-yönet taktikleriyle ele alınmamalıdır. Öğrenen sistemlerdeki diğer bir sorun olan tümevarımsal yanlılık seçiminin genelliği desteklemek yönünde olması ve üstuydurmadan (overfitting) kaçınması, seyrek sınıflar/durumlar için başarımı olumsuz etkileyebilir. Başlı başına bir sorun olan *gürültü* olgusu da, seyreklik içeren problemlerdeki öğrenme üstünde daha belirgin etki gösteriir. Az sayıdaki gürültülü örnek öğrenilen altkavramları etkiler ve üstuydurmalardan kaçınmanın daha önemli olduğu görülür.

Seyreklik (sınıf dengesizliği) problemine daha önce sunulan çözümleri kategorilere ayırmak bağlamında, araştırmacılar farklı görüşler ortaya koymaktadır. Bir görüşe göre [18], çözümler üç grupta incelenebilir: 1) veri önişlemeye dayalı, 2) algoritmaya dayalı ve 3) öznitelik seçimine dayalı. Maheshwari ve diğerleri [19], yöntemleri iki ana gruba ayırmayı seçerek yeni algoritma tasarımına ve farklı algoritmalar kullanmaya dayalı çözümleri *iç yaklaşımlar*, veri dengesizliğinin etkisini yok etmek için veriyi önceden işlemeyi *dış iş süreci* olarak adlandırmıştır. Diğer bir gruplama [20]; veri kümesinin değiştirilerek dengeli dağılım gösteren duruma getirildiği *veri-düzeyinde* yöntemler, tek bir sınıflandırıcı veya kolektif sınıflandırıcılardan oluşup seyreklik problemini amaca yönelik tasarlanan *algoritmalar*la çözmeye çalışan yaklaşımlar ve hem veri hem algoritmaların işin içinde olduğu *karma çözümler* olarak yapılmaktadır. Birçok çözüm hem veriyi dengeleme hem de seyrekliği ele alan algoritmalar kullanma yollarına beraber başvurduğu için karma çözümler kategorisindedir.

Bu bölümün kalanında, seyreklik problemine farklı bakışlar ve önerilen çözümlere, kategorileştirmenin tam olarak nasıl yapıldığına takılmaksızın, örnekler sunulmaktadır. Belli bir başlıkta sunulan bir yöntem, başka yönlerine bakılırsa diğer bir başlıkta sunulmaya uygun olabilir. Açıklamaların çoğu seyrek sınıflara yönelik olmakla beraber, seyrek durumlar için söylenecekler de benzerdir ve çevrim doğrudan yapılabilir.

4.1. Uygun Değerlendirme Ölçevleri

Seyrekliğin üstesinden gelmeye çalışan yöntemler, genellikle seyrekliğin yol açtığı belirli sorun veya sorunlara odaklanarak istenmeyen etkilerin azaltılması temelinde iş görürler. Bu bağlamda, seyrek veriyle çalışmak zorunda olan sistemlerin başarım ölçevlerinin ne olduğuna uygun bir biçimde karar verilmesi ilk gereksinimdir. 2. ve 3. bölümlerdeki ölçevlerin ve başarım değerlendirme araçlarının doğru kullanımı [8][9][21], tasarımcıların dikkatle üstünde durması gereken işlerdir. Geri getirme ve kesinlik ölçevlerinin çeşitlemeleriyle veri madenciliği sürecini yönlendiren ve alınan sonuçları değerlendiren sistemler [1][22][23][24], genetik algoritmaları koşan ve her yinelemeden sonra evrilen sınıflandırma kurallarının kullanılırlığını ölçmek için F-ölçevi kullanan [22], farklı veri madenciliği algoritmalarının başarımlarını karşılaştırmak için yine F-ölçevinden yararlanan [25][26][27] çalışmalar bulunmaktadır. Quinlan [28], sınıf önsel olasılıklarını (sınıf dağılımlarını) dikkate alan, böylece seyrek sınıfların ve küçük bölgelerin doğruluk kestirimlerini iyileştiren daha gelişmiş bir hata kestirim ölçevi sunmaktadır. Di Martino ve diğerleri [29], en çok doğruluk yerine en çok F-ölçevi değerini, dengesiz dağılımlı problemlerde çalışması hedeflenen bir sınıflandırıcı tasarımında kullanmaktadır.

4.2. Tümevarımsal Yanlılık

Tümevarımsal yanlılık, öğrenmenin gerçekleşebilmesi için özgül örneklerden yola çıkarak genelleştirme yapmak üzere olmazsa-olmaz bir unsurdur. Veri madenciliği sistemlerinin çoğunun tümevarımsal yanlılık tercihi, bütüncül sınıflandırma başarımını artırmak için özelleştirme yerine genelleştirmeden yanadır. Bu tercih, yaygın durumlar için iyi olmakla beraber, seyrek durumlar için uygun değildir ve bazen tamamen yok sayılmalarına yol açabilir. Oysa ki, ilginç nesneler seyrek olanlardır ve bu veri kümeleri için de geçerlidir. Seyrek durumlar/sınıflar üzerinde başarımı artırmak için yanlılık tercihini değiştiren yaklaşımların bazıları [30] bu amaçlarına ulaşırken bütüncül başarımı azaltmakta, çoğu karma yaklaşımlarla iş gören bazıları da [31][32] karışık başarım gözlenen (bazen daha iyi bazen daha kötü olarak değerlendirilen) sonuçlar üretmektedir.

4.3. Uygun Arama Yöntemleri

Sınıflandırma kurallarını oluştururken kullanılan *fırsatçı* (greedy) arama algoritmaları, seyreklik söz konusu olduğunda başarısız olabilir. *Fırsatçı olmayan* (nongreedy) arama yaklaşımlarından biri genetik algoritmalar kullanmaktır. Tek bir çözüm yerine, aday çözümlerden oluşan populasyonlar ile çalışan ve global arama teknikleri olan genetik algoritmalar [33], arama sürecini yönlendirmede rasgele (stokastik) işleçler kullanırlar. Bu özellikleriyle, özellik etkileşimlerini daha iyi ele alarak yerel maksimumlarda tıkanmaktan kaçınmayı sağlayan genetik algoritmalar seyreklik problemini gidermede uygun yöntemlerdir [22][23][34]. Fırsatçı olmayan aramaya bir diğer örnek olan ve kaba kuvvet arama gerçekleştiren *Brute* [35], aramayı genellikle fırsatçı algoritmalarla yapan karar ağacı öğrenme yöntemlerine alternatif olarak geliştirilmiştir. Brute, doğru olan birleşimli kurallara varmak amacıyla ayrıntılı bir derinlik-sınırlı arama yapmaktadır. Genelde ayrıntılı arama yöntemleri kullanan ortaklık-kuralı madenciliği sistemleri de [36], kuramsal olarak seyrek ortaklıkları bulma yeteneğine sahiptir.

4.4. Maliyete-duyarlı Öğrenme

Seyrek sınıflar/durumlar bağlamında sınıflandırıcı başarımının artırılması için maliyete-duyarlı veri madenciliği yapmak başka bir alternatiftir. Tıbbi tanı sistemleri gibi birçok sınıflandırma işinde, birincil önemi olan sınıflar/durumlar seyrek olanlardır. Böyle problemlerde, yaklaşım, seyrek olan (pozitif) sınıfın yanlış sınıflandırılmasına, yaygın olan (negatif) sınıfın yanlış sınıflandırılmasına göre daha fazla maliyet atamak (ceza kesmek) ve böylece başarımı artırmak olabilir. Belirli maliyet bilgisini bulmak genelde zor olmakla birlikte, alan bilgisi olan uzmanlardan yardım almak bir seçenektir. Yanlış negatif (*FN*) ve yanlış pozitif (*FP*) öngörüler arasında olabildiğince doğru bir oran bulmak için bu oranı bir amaç fonksiyonunun yeterli bir değeri elde edilene kadar değiştirme [15], çok-sınıflı problemlerde herbir sınıfın en uygun yanlış-sınıflandırma maliyetini bulmak için genetik algoritmalar kullanma [37] ve maliyet/yarar çözümlemesi yaparak seyrek olaylarda sınıflandırma için koşulan *STOCS* (Statistical Online Cost Sensitive Classification) [38] bu kategoridendir. Frumosu ve diğerleri [39], maliyete-duyarlı öğrenmeyi üretim mühendisliğindeki ürün kusurlarının öngörüsünde kullanmaktadır.

4.5. Örnekleme

Mutlak anlamda az sayıda durum/sınıf örneği içeren veya birden çok durum/sınıf örneği arasında birinin diğerlerine göre az örnek barındırdığı (görece seyreklik) öğrenme kümelerinde, seyreklik sorunlarının üstesinden gelmek için öğrenmede kullanılan verinin dağılımını değiştiren süreçler uygulamak tercih edilen yollardan biridir. Beklenti, bu türden bir önişlemenin, sınıflandırıcıların pozitif (seyrek) sınıf üyelerini saptama başarımını artıracağı ve beraberinde gerçekçi (akla uygun) nicel bir başarım ölçüsü üreteceğidir.

Öğrenme kümelerindeki veri dağılımı (seyreklik/dengesizlik) nedeniyle oluşan sorunları ele almak için kullanılan en temel dağılım değiştirme yöntemi *örneklemedir* (sampling). Basit örnekleme teknikleri, çoğunluk sınıfına ait örnekleri rasgele seçerek çöpe atan *altörnekleme* (undersampling) ve seyrek sınıf örneklerini rasgele seçerek kopyalarını oluşturan *üstörneklemedir* (oversampling). Rasgele altörnekleme değerli bilgiyi kaybetmeye neden olabilir, rasgele üstörnekleme de öğrenim kümesine seyrek sınıf örneklerinin birebir kopyalarını ekleyip yeni bir bilgi katmaz ve üstuydurma riskini artırır.

Örneklemenin temel biçimleri pratikte yeterince iyi sonuçlar vermediği için, bazı buluşsal (heuristic) örnekleme yöntemleri ortaya çıkmıştır. Özel çoğunluk sınıfı örneklerini (gürültülü, artık veya iki sınıfı ayıran sınıra yakın) eleyip bütün seyrek sınıf örneklerini tutan *OneSidedSelection* [40] ve herbir örneği ve k en-yakın komşularını kullanarak yapay (sentetik) örnekler üreterek seyrek sınıf üyelerinin üstörneklemesini yapan *SMOTE* (Synthetic Minority Oversampling Technique) [41] bu türden yöntemlerdir. *Borderline-SMOTE* [42], bütün seyrek sınıf örneklerini kullanmak yerine, yalnızca yanlış sınıflandırılmaya daha yatkın olan seyrek sınıf örneklerini kullanarak yapay örnekler elde eder. Yanlış sınıflandırılmaya yatkın olan seyrek sınıf örnekleri, iki sınıfı ayıran sınıra yakın konumda bulunanlardır. Bir seyrek sınıf örneğinin sınıra yakın olup olmadığının kararı, en yakın birkaç komşusunun sayılarak kaçının çoğunluk sınıfına kaçının seyrek sınıfa ait olduğunun gözlemlenmesine dayalı verilir.

Garcia ve diğerleri [43], evrimsel-tabanlı yöntemleri de kullanarak sınıflandırılması gereken örneklerin *Öklid n-uzayı*nda tutulan genelleştirilmiş örneklere olan uzaklığına göre sınıflandırıldığı bir yaklaşım sunmaktadır. Evrimsel yaklaşım, en uygun genelleştirilmiş örneklerin seçiminde bir eniyileme aracı olarak kullanılır. Das ve diğerleri [44], var olan üstörnekleme yöntemleri kullanılıp yeni yapay örnekler üretilirken, seyrek sınıfın olasılık dağılımının genelde dikkate alınmadığından yola çıkarak *RACOG* ve *wRACOG* adlı iki olasılıksal üstörnekleme yaklaşımı sunmaktadır. Önerilen yöntemler, yeni seyrek sınıf örneklerini üretip hangilerinin seçilmesi gerektiğine karar verirken, özniteliklerin birleşik olasılık dağılımından ve Gibbs örneklemeden [45] yararlanır. *RACOG*'da Gibbs örnekleyicinin önceden tanımlı bir gecikme değerine bağlı ürettiği, *wRACOG*'da öğrenme modelinin yanlış sınıflandırma olasılığının en yüksek olduğu örnekler seçilmektedir. Susan ve Kumar [46], dengesiz veri kümelerinin örneklenmesi konusunda eniyileme için çoğunluk ve azınlık sınıflarından örneklerin akıllı gösterimler ile seçilmesine dayalı tekniklerle ilgili bir derleme sunmaktadır.

4.6. İteleme

Temel olarak öğrenme kümelerinin dağılımlarını değiştiren ve genelleştirilmiş bir örnekleme yöntemi olarak görülen *iteleme* (boosting), zayıf taban öğrenicilerin başarımını artırabilen bir kolektif öğrenme yöntemidir. Bir dizi temel sınıflandırıcı ele alındığında, öğrenme örneklerinin ağırlıkları uyarlamalı olarak değiştirilir ve önceki yinelemede yanlış sınıflandırılan örneklerin ağırlığına diğer örneklerinkinden daha büyük atamalar yapılır. Seyrek sınıf örnekleri çoğunluk sınıf örneklerine göre hataya daha yatkın oldukları için, itelemenin seyrek sınıf örneklerine daha büyük ağırlıklar atayıp sınıflandırma başarımlarını artıracağına inanmak akla uygundur. Standart itelemede [47], doğru pozitif (*TP*) ve doğru negatif (*TN*) örneklere eşit önem verilir ve bu yüzden art arda yinelemelerden sonra öğrenim kümesindeki çoğunluk sınıfı baskın olmayı sürdürebilir. *RareBoost* (seyrek iteleme) [48], hem kesinlik hem geri çağırma ölçevlerine odaklanarak pozitif ve negatif örneklerin ağırlıklarını farklı değiştirir. *SMOTEBoost* [49] yaklaşımında, iteleme yinelemelerinde *SMOTE* [41] uygulanır. *AdaBoost* [47] yönteminin değişik bir türü olan *AdaCost* [50], maliyete-

duyarlı bir teknik benimseyerek birikimli yanlış sınıflandırma maliyetini azaltmak için öğrenim kümesinin dağılımını değiştirirken iki tip hataya farklı maliyetler atar. İtelemenin seyrekliğe bir çözüm sunmasının yanı sıra, taban sınıflandırıcının niteliklerinin de başarımda ciddi etkiye sahip olduğunu gösteren çalışmalar [27] ve çok-sınıflı problemlerde iteleme kullanımıyla ilgili derlemeler [51] de bulunmaktadır.

4.7. Kolektif Öğrenme Yöntemleri

Tek başına iş gören sınıflandırıcıların başarımını artırmak amacıyla birden çok sınıflandırıcının öğrendiği ve alınan yanıtların birleştirildiği düzenekler *kolektif* (ensemble) yöntemler olarak bilinir. Birleştirme, genelde iteleme veya *önyükleme toplama* (Bootstrap AGgregation: bagging) [52] uygulanarak yapılmaktadır. İtelemede, söz konusu sınıflandırıcıların herbiri bir önceki sınıflandırıcının hatalarına odaklanarak üretilmekte ve yeni sınıflandırıcının hatalardan kaçınması için ağırlıklar uygun olarak belirlenmektedir. Önyükleme toplamada, öğrenme kümesi D'den birbiçim ve yerine koyarak örnekleme ile m tane öğrenme kümesi oluşturulur ve herbiri kullanılarak m model elde edilir. Kolektif sınıflandırıcıların kararları m tane kararın eşit ağırlıkta oylanmasıyla verilir. Önyükleme toplamanın yaklaşımı, eldeki veriden çok sayıda öğrenim kümesi üreterek öngörü değişintisini azaltmak odaklıdır.

Guo ve Viktor tarafından geliştirilen *DataBoost-IM* yönteminde [53], iteleme algoritmasının koşması sırasında, hem çoğunluk sınıfı hem de seyrek sınıftan sınıflandırılması zor örnekler belirlenmektedir. Sonra, zor örneklerden her iki sınıf için ayrı ayrı yapay örnekler üretilmekte ve öğrenme kümesine eklenmektedir. Böylelikle, yeni öğrenme kümesindeki sınıf dağılımları ve farklı sınıfların toplam ağırlıkları dengelenir. Yapılan denemelerde, yöntemin bir sınıfı diğerine göre gözden çıkarmadığı ve iki sınıf için de yüksek doğruluklu öngörüler yaptığı gözlemlenmektedir.

Kang ve Cho [54], *altörneklenen destek vektör makineleri topluluğu* (Ensemble of Under-Sampled SVMs: *EUS SVM*s) adını verdikleri yöntemle, özellikle seyrek sınıfa ait örnek sayısının az olduğu durumda diğer yöntemlere üstün gelindiğini göstermektedir. Bu yöntem, destek vektör makinelerinin genelleştirme yeteneğini iteleme kullanarak kolektif öğrenme kapsamında bir araya getirmekte, altörneklemenin eksikliğini ortadan kaldırmakta ve üstörneklemenin zaman karmaşıklığını azaltmaktadır.

Liu ve diğerlerinin *EasyEnsemble* yönteminde [55], negatif sınıftan C tane ve aynı sayıda eleman içeren örnek kümeleri elde edilmektedir. Örnek kümelerinin herbirindeki eleman sayısı n pozitif sınıfın tüm elemanlarının sayısıyla aynıdır. Daha sonra, pozitif örnekler ve negatiflerden elde edilen örnek kümelerinin herbiri ayrı ayrı bir araya konmakta ve sınıflandırıcı *AdaBoost* [47] ile öğrenmektedir. En sonunda öğrenen bütün *AdaBoost* toplulukları, önyükleme toplama ile birleştirilmektedir. *EasyEnsemble*'daki temel motivasyon, altörneklemenin verimliliğinden yararlanırken potansiyel olarak yararlı olabilecek bilgi kaybını engellemektir. Aynı çalışmada sunulan *BalanceCascade* [55] yöntemi yine *AdaBoost* topluluklarını kullanarak çalışmakta ama negatif örnekleri rasgele silmek yerine bunu bir yönlendirilmiş silme yaklaşımıyla yapmaktadır. *EasyEnsemble* yöntemindeki sınıflandırıcılar paralel biçimde öğrenir, buna karşın *BalanceCascade*'de öğrenme sıralıdır ve bir sınıflandırıcının doğru sınıflandırdığı örnekler, sonraki sınıflandırıcıların öğrenim kümelerinde kullanılmaz.

Guo ve diğerlerinin [56] yaklaşımında, her taban sınıflandırıcının iki adımda oluşturulduğu, kısıt izdüşüm (constraint projection) ve altörneklemeden yararlanan bir kolektif öğrenme yöntemi sunulmaktadır. İlk adımda, negatif ve pozitif sınıf kümelerinde altörnekleme yapılarak ikili kısıtlardan oluşan bir kısıtlar kümesi oluşturulmakta ve bu kümeden bir izdüşüm matrisi öğrenilmektedir. İkinci adımda, ilk öğrenme kümesi altörneklenmekte ve edinilen yeni öğrenme kümesi kullanılarak, izdüşüm matrisiyle tanımlanan yeni öznitelik uzayında bir taban sınıflandırıcı elde edilmektedir. İlk adım temel sınıflandırıcıların çeşitlemeleri olmasını amaçlamakta, ikinci adım temel sınıflandırıcıların seyrek sınıf örneklerinin sınıflandırma başarımını artırmayı ve çeşitlemeyi daha da ileri götürmeyi hedeflemektedir.

4.8. Öznitelik Seçimi

Çok boyutlu verideki ilgisiz öznitelikler, özellikle dengesiz dağılımlı öğrenme kümeleri olan problemlerde sınıflandırma başarımını düşürür [57][58] veya kayda değer biçimde artırmaz [59]. Üstelik, çok öznitelik kullanımı öğrenme ve tümevarım süreçlerini yavaşlatır. Bu durumda, sınıflandırma başarımını artıran önemli öznitelikleri seçmek amaçlanmalıdır. Temel öznitelik seçimi yöntemleri arasında ilinti katsayısı [60], ki-kare [61][62], olasılıklar oranı (odds ratio) [2][63] ve bilgi kazanımının (information gain) [64] kullanımı sayılabilir.

Kira ve Rendell [65][66], herbir öznitelik için puan hesaplayan ve bu puanları sıralayıp seçimde en yüksek puanlı öznitelikleri kullanan bir yöntem sunmaktadır. Zheng ve diğerleri [67], metin sınıflandırmada çok boyutlu dengesiz veri kümeleri için uygun olan bir bağlamda, pozitif ve negatif sınıfların özniteliklerini kullanarak özel bir biçimde birleştiren bir öznitelik seçimi çerçevesi sunmaktadır. Ertekin ve diğerleri [68], dengesiz dağılımlı veri özelliği bulunduran ağdaki (web) metinlerin kategorizasyonunda farklı öznitelik seçimi tercihleri için başarımı incelemektedir. Chen ve Wasikowski [69], sınıflandırma başarımını ölçmek için doğruluk yerine ROC-tabanlı öznitelik seçimi kullanmaktadır. Alibeigi ve diğerleri [70], özniteliklerin bütün sınıflar üzerindeki olasılık dağılımlarını ve ilintilerini ele alarak, her özniteliğin sınıflandırmadaki katkısını derecelendirmektedir. Elde edilen dereceler, öznitelik seçimini yönlendirmektedir. Yin ve diğerleri [71], iki yeni öznitelik seçimi yaklaşımı sunmaktadır. İlkinde, büyük sınıflar görece daha küçük sözde altsınıflara bölünüp etiketlenmektedir. İkincisinde, hesaplanmasında sınıfların önsel olasılık bilgisini kullanmadığı için güçlü derecede kayıklık-duyarsız (skew-insensitive) ve bir dağılım ıraksaması ölçüsü olarak işe yarayan Hellinger uzaklığını [72] kullanan bir yöntem önerilmektedir. Deneysel sonuçlar, dengesiz veri üzerinde diğer öznitelik seçimi yöntemlerine üstünlük göstermektedir. Jovic ve diğerleri [73], standart süzgeç (filter), örtü/sarıcı (wrapper) ve gömülü yöntemlerle beraber karma öznitelik seçimine bir bakış sağlamaktadır.

4.9. Kural-tabanlı Yöntemler

Tümevarımsal yöntemler, seyrek sınıflar içeren problemlerde kesinliği ve geri çağırmayı enbüyütmeye çalışmaktadır. Birbiriyle yarışan iki ölçevi birlikte büyütmek karmaşık seyrek sınıf problemlerindeki dağınık yanlış pozitifler ve seyrek pozitif örneklerden kaynaklanan küçük

bölgeler gibi nedenlerle çok zordur. Bu sorunların üstesinden gelmek için *PNRule* [26] iki-aşamalı *kural-tümevarımı* (rule-induction) yaklaşımı sunmaktadır. İlk aşamada, hem pozitifler hem de negatifler içerebilecek yüksek destek ve makul doğruluk değerine sahip kurallar bulunur. İkinci aşamada, doğruluğu artırmak için yanlış pozitifleri ortadan kaldıran kurallar geliştirilir. Başka bir deyişle, ilk aşamada geri çağırmaya odaklanılır, ikinci aşamada kesinlik eniyilenir. *PNRule* özellikle seyrek sınıf problemlerinde uygulanmaya elverişlidir.

Görülmeye başlanan örüntüler (Emerging Patterns: *EP*) [74], bir sınıf içindeki destekleri diğer sınıflardakilerden önemli derecede fazla olan öğe kümelerini gösteren ve sınıflar arasındaki önemli çoklu-özellik farklılıklarını yakalayabilen, daha yeni bir örüntüler türüdür. Görülmeye başlanan örüntülerin (*EP*) seyrek sınıflar üzerindeki ayırıcı gücünü kullanan ilk yaklaşım olan *EPRC* [75] işini üç aşamada yapar. Önce, bulunmamış yeni seyrek sınıf *EP*leri üretilir, sonra düşük destekli *EP*ler budanır, sonra da seyrek sınıf *EP*lerinin destekleri artırılır. *EPDT* [76] ve *DEP* [77], seyrek sınıf problemleri için geliştirilen diğer *EP* yöntemleridir.

Bulanık kural-tabanlı sınıflandırma sistemleriyle ilgili başka bir çalışmada [78], parametrik birleşim işleçleri kullanılarak işleyen uyarlamalı bir çıkarım sistemi aracılığıyla, bu sistemlerin dengesiz verideki davranışı incelenmektedir. Verideki dağılımı dengelemek için *SMOTE* [41] ile bir önişleme de uygulanmaktadır. Görgül (ampirik) sonuçlar, kullanılan parametrik birleşim işleçleriyle dengesizlik oranları farklı tüm kümelerde daha yüksek başarım göstermektedir.

4.10. Tanıma Yöntemleri (1-Sınıf Sınıflandırma)

Bütün sınıfları ayırt eden sınıflandırma kurallarını aramak yerine yalnızca seyrek sınıfı belirleyen bir tanıma yaklaşımı kullanılabilir. Tek sınıfı tanımak üzere sinir ağlarıyla yalnızca pozitif örneklerden öğrenen *Hippo* [79] ve aynı amaçla *destek vektör makineleri*nin kullanımının [80] yanı sıra, öğrenmeyi gerçekleştirirken bütün sınıfların örneklerini kullanıp yalnızca seyrek sınıfı öğrenen Shrink [1], Ripper [81], Brute [35] gibi çalışmalar bulunmaktadır. Japkowicz [82], dengesiz dağılımlı veri üzerinde 1-sınıf sınıflandırıcıların başarımını ikili sınıflandırıcılarınki ile karşılaştırmakta, örnek verinin hem örneklemeli hem örneklemesiz çeşitlemelerini kullanarak örneklemeli ikili sınıflandırıcının üstünlüğü sonucuna varmaktadır. Buna karşın, sonraki çalışmalar [83][84], önceki bulguyla çelişmekte ve yüksek derecede dengesiz olan veride 1-sınıf sınıflandırıcıların tercih-edilirliğini bildirmektedir.

4.11. Veriyi veya Problemi Bölme

Çoğunluktaki sınıfın altsınıflara bölünerek problemdeki seyreklik derecesini azaltmak yönünde çalışan yöntemler, bu doğrultuda gruplandırma gerçekleştirip karmaşık kavramları daha küçük parçalara ayırarak seyrekliğin üstesinden gelmektedir. Gruplamadan anlaşıldığı üzere, böyle yöntemler, hem öğreticisiz hem öğreticili sınıflandırma için kullanılabilir [85]. Başka bir çalışmada, çoğunluk sınıfının yerel gruplama ile altsınıflara bölünmesi, seyrek sınıfın üstörneklenmesi ile birleştirilmektedir [86]. Amaç, seyrek sınıfın ortalama büyüklüğünü bölünen çoğunluk sınıfının parçalarının ortalama büyüklüğüne yaklaştırmak ve seyrekliği dikkate alınması gereken bir etken olmaktan uzak tutmaktır. Başka benzer yaklaşımlar, veriyi gruplama ile işleyip sınıf sayısını artırmak yerine öğrenme kümesini bölgelere ayırarak bütün örnek uzayında seyrek olmakla beraber, ayrılan bölge içerisinde veya altproblemlerde seyrek olmayan sınıflarla çalışmaktadır [87].

4.12. Algoritmik Çeşitlemeler

Ortaklık kurallarını (association rules) bulan sistemlerde, seyrek görülen öğeler arasındaki ve aslında güçlü ortaklıkların belirlenmesi, en az destek değerinin bulunan ortaklıkların sayısında ele alınması zor olan patlamalara yol açmaması için düşük alınmaması gereğiyle, klasik *Apriori* yöntemi [88], sözü edilen ortaklıkların bulunmasında başarılı olamamakta ve bir değişiklik veya ekleme gerekmektedir. Böyle bir durumda, az görülen öğeler için destek değerini düşük almak işe yaramaktadır. Bütün öğeler için bakıldığında, herbir öğe için birlikteliklerde görülme sayılarına göre farklı destek değerleri belirlenmekte ve bir ortaklık kuralına karar verilirken, birliktelikte görülen öğelerden küçük destek değerine sahip olanınki dikkate alınmaktadır. Sonuç olarak, özellikle seyrek öğeler farklı biçimde ele alınarak önemli ortaklık kuralları belirlenebilir [89]. Seyrek öğeleri ele almak odaklı *Apriori* algoritmasının çeşitlemeleri arasında; en az destek yerine en çok destek değerini kullanarak bu değerden küçük desteği olan aday öğe kümelerini bulup daha sonra ortaklık kurallarını bilinen *Apriori* ile üreten *AprioriInfrequent*, bazı eklentilerle beraber yine en çok destek değerini kullanan *AprioriInverse* [90] ve sık öğeleri belirleyip bilinen yöntemle ortaklıkları bulan ama seyrek öğeleri listeledikten sonra bu öğeleri başka bir algoritmayla işleyen *AprioriRare* [91] sayılabilir.

Batuwita ve Palade [92], aykırı değerler ve gürültü problemini ele almak için kullanılan *bulanık destek vektör makineleri* (Fuzzy Support Vector machines: *FSVM*) sınıflandırıcısını, seyrek sınıfların varlığında da çalışması için geliştirmektedir. Önerilen yöntemde, hem aykırı değerler ve gürültülü veri probleminin hem de seyrek sınıf probleminin etkisini azaltmak üzere maliyete-duyarlı öğrenme ilkesi gözetilerek örneklere bulanık üyelik değerleri atanmaktadır.

4.13. Alan Bilgisinin Etkileşimli Kullanımı

Doğasında etkileşimli bir süreç olan veri madenciliğinde insanların (uzmanlar) alan bilgisinden yararlanmak, seyreklik içeren problemlerde, başarımı artırmaktadır. Bilginin kullanımı, daha ileri öznitelikler sağlayarak örneklerin daha iyi tanımlanması ve öznitelikler arasındaki ilişkilerin problem açısından ne yönde yararlı olacağının ortaya konması yönünde işe yarayabilir. Alan bilgisi, seyrek sınıfları öngörmede en yararlı öznitelikleri önerme eğilimindedir. Kimi problemlerde, elde edilen sonuçlardan hangilerinin ilginç olduğu kararının uzmanlarca verilmesi, bu sonuçlar üzerinde ek madencilik yapılarak seyrek sınıfların aranmasında yardımcı olmaktadır [93]. Kopanas ve diğerleri [94], bir telekomünikasyon şirketinin müşterilerinin borçlarını ödeyememe profilleri üzerinde tanımlanan bir sınıflandırma problemi için alan bilgisinin nasıl kullanılabileceğine ilişkin bir örnek sunmaktadır.

Uzmanlar; öğrenme kümesindeki müşterilerin ilgisiz özelliklerini eleme, bazı birden çok öncelikli değerden daha soyut özellik çıkarma, eksik değerleri belirleme, gözlem yapılan dönemleri ve zaman ölçeğini tanımlama, örnekleme ve işlem eleme yollarıyla veri azaltmada yarar sağlamaktadır.

4.14. Diğer Yöntemler

Peréz-Godoy ve diğerleri [95], dengesiz dağılımlı veriyle öğrenen sınıflandırma problemlerine uygulanan radyal taban fonksiyonu ağlarının tasarımında, evrimsel işbirlikli-yarışmalı model (CO^2RBFN) kullanmakta ve *SMOTE* [41] ile işbirliğini incelemektedir. Önerilen yöntem; dengesiz dağılımlı örnekler üzerinde, gösterici nitelikteki sinir ağları, C4.5 karar ağacı [96] ve sıradüzensel bulanık kural-tabanlı sınıflandırıcılarla karşılaştırılmakta ve başarımın iyi olduğu gösterilmektedir. Nguwi ve Cho [97], dengesiz dağılımlı veride başarım sağlamak amacıyla, öznitelik seçimini ve sınıflandırma algoritmasını beraber kullanan karma bir yaklaşım sunmaktadır. Öznitelik seçimindeki ölçüt destek vektör makinelerinden elde edilmekte ve özniteliklere bağlı ağırlık vektörü duyarlılığına dayanmaktadır. Sınıflandırmada yararlanılan araç olarak, öğreticisiz *görülmeye başlanan kendini örgütleyen eşlem* (Emergent Self-Organizing Map: *ESOM*) [98] kullanılmaktadır. *ESOM*, çok-boyutlu verinin görülmeye başlanan esas yapısal özelliklerini iki-boyutlu bir eşlemde gösterebilmeyi sağlamaktadır. Haines ve Xiang [99], seyrek sınıfların bilinmediği ve öğrenme sırasında bulunması gereken problemler için, bir aktif öğrenme yöntemi önermektedir. Yöntemde, verilen bir örnek kümesi için, etiketlenmesi gereken örnekler aktif öğrenme ile seçilmektedir. Amaç, seçilen örneklerle bir sınıflandırıcı oluşturmak, bunu yaparken en iyi sınıflandırıcıyı elde etmekle harcanan çaba miktarı arasındaki dengeyi gözetmek, yapılan tercihlerle bu dengeyi eniyilemektir. Problem, hem seyrek sınıfları bulma hem de sınıflandırma yapmaya beraber odaklanmaktadır. Wankhade ve diğerleri [100]; *k*-ortalamalar, kolektif öğrenme ve bölme-birleştirme yöntemlerini kullanarak sınıflandırma ve gruplamaya dayanan karma bir yaklaşım sunmaktadır. Koziarski ve diğerleri [101], etiket (sınıf kimliği) gürültüsünün olduğu çok-sınıflı dengesiz dağılımlı problemler için ayıklama (temizleme) ve yeniden örneklemenin birlikte kulanıldığı bir algoritma göstermektedir.

Birçok yönden incelenen ve çözüm arayışları süregelmekte olan seyreklik ve sınıf dengesizliği problemleri için, veri madenciliği disiplini derin öğrenme bakış açısıyla ele alındığında, ortaya konan yöntemlere ilişkin bir derleme Johnson ve Koshgoftaar [102] tarafından sunulmaktadır. Aynı problemlere büyük veri çerçevesinden ayrıntılı bir inceleme Leevy ve diğerlerinin [103] çalışmasında bulunabilir.

5. Sonuç

Veri madenciliğindeki karar verme süreçlerinde, dengesiz durum/sınıf dağılımları söz konusu olduğunda, sınıflandırma başarımının mantıklı ve doyurucu bir nicel ölçüsünün ortaya konması uzunca zamandır üzerinde durulmakta olan bir araştırma konusudur. Tablo 2, seyreklik içeren üç örnek veri kümesine ilişkin bilgi göstermektedir.

Tablo 2: Dengesiz dağılımlı veri kümelerinin örnekleri

VERİ KÜMESİ	*#p*	*#n*	*n* (%)
Medicare Part B [104]	1,409	3,691,146	99.962
PubChem AID 373 [105]	62	59,726	99.896
Fetal Kafatasları [16][17]	29	329	91.8994

Değerlendirmeler ve görüşler, başarım ölçme konusunun çözülmek istenen problemin doğası ve ne başarılmak istendiğiyle yakından ilgili olduğunda birleşmektedir. Bununla beraber, seyrek durum/sınıf içeren bütün problemler için ortak bir çerçeveden bakıldığında, seyrek sınıfın öngörülmesinde geri getirme ve kesinlik değerlerinin ikisinin de daha yüksek olması ve sınıflandırıcıları yüksek değerler doğrultusunda ödüllendirme [21], uygun ölçevi belirlemedeki ana amaçtır.

Sınıflandırıcı başarımını görselleştirmek ve ölçmek için yararlı olduğu gösterilmiş bir diğer araç da ROC çizgeleridir [8]. Doğruluk, hata yüzdesi, yanlışların maliyeti gibi noktasal başarım ölçüleriyle karşılaştırıldığında, ROC çizgelerinin daha kapsamlı bir başarım ölçüsü sunduğu görülür. Sınıf kayıklığı (dengesiz sınıf dağılımları) ve hata maliyeti unsurlarını birbirinden ayrıştırdığı için, kesinlik-geri getirme eğrileri gibi diğer çizgesel başarım gösterimi araçlarına göre üstün olan ROC çizgeleri, yine de bilinçli kullanılmalıdır.

Bir sınıflandırıcıyı gerçekleyip örnek veri üstünde başarımını ölçmeden önce asıl yapılması gereken, sözü edilen başarım ölçme konusunda problemlere yol açan seyreklik veya dengesiz durum/sınıf dağılımı görüngüsünün üstesinden gelmek ve tasarımda çözümler bulabilmektir. Bu amaçla, literatürde önerilen bir çok yaklaşım bulumaktadır. Öneriler, öğrenme verisinin önişlemeden geçirilerek dengesiz sınıf dağılımlarının daha dengeli duruma getirilmesi olabileceği gibi, sınıflandırma algoritmalarının dengesiz veriyle iş görecek biçimde tasarlanmasıyla da olabilir. Bu yolların izlenmesinde izlenecek çeşitlemeler çok sayıda olup, neredeyse her zaman, iki yaklaşımdan da yararlanan karma yöntemlerden söz edilebilir. Verinin özniteliklerinin doğru seçimi ve problemin çözümüne en yararlı olacak biçimde düzenlenmesi de bu bağlamda ayrı bir yaklaşım olarak görülebilir.

Başarımı değerlendirmede kullanılan uygun ölçevlerin seçimi, başlı başına seyreklik problemiyle baş etmek için bir çözüm olabilirken, sınıflandırıcıların tümevarımsal yanlılık seçiminin farklılaştırılması da öngörülerin daha amaca uygun biçimde gerçekleşmesine olanak tanıyabilir. Sınıflandırma kurallarını oluştururken öğrenme kümesi verisi üzerinde uygun arama yöntemlerini kullanmak, tasarım sırasında maliyete-duyarlı öğrenme yaklaşımından yararlanıp yanlış kararlar için uygun cezalar belirlemek, örnekleme gerçekleştirerek veri dağılımını değiştirmek, temelinde yine bir tür örnekleme olup zayıf taban öğrenicileri yinelemelerle güçlü duruma getiren itelemeden yararlanmak, karar verici olan tekli sınıflandırıcılar yerine kolektif öğrenme ile birden çok sınıflandırıcının öğrenmesini sağlayarak kararı bir tür oylama ile aldırmak, tanıma yöntemlerine başvurmak ve yalnızca bir sınıfın varlığını aramak, problemi/veriyi bölme yaklaşımlarına gitmek, uzman bilgisini etkileşimli olarak kullanmak gibi birçok yol seyreklik içeren problemlerde kullanılmaktadır. Yaklaşımları gruplamak için başka yollar izlenebileceği gibi, adı geçen belirli örnek yöntemlerin başka kategorilerde bulunması gerektiğini savunan görüşler de olabilmektedir. Tablo 3, sözü edilen çözümler ve kaynaklara ilişkin bir özet sunmaktadır.

Tablo 3: Seyreklik problemini gidermek için çözüm önerileri

Uygun değerlendirme ölçevlerini seçme [8-9][21-29]
Tümevarımsal yanlılık tercihini değiştirme [30-32]
Uygun arama yöntemlerini kullanma [22-23][33-36]
Maliyete-duyarlı öğrenme gerçekleştirme [15][37-39]
Örnekleme ile veri dağılımını değiştirme [40-46]
İteleme (genelleştirilmiş örnekleme) kullanma [47-51]
Kolektif yöntemler [52-56]
Öznitelik seçimi ile ilgisiz öznitelikleri eleme [2][57-73]
Kural-tabanlı yöntemler [26][74-78]
Tanıma (1-sınıf sınıflandırma) [1][35][79-84]
Veriyi veya problemi bölme [85-87]
Algoritmaları çeşitlendirme [88-92]
Alan bilgisini etkileşimli kullanma [93-94]
Olası diğer yaklaşımlar [95-101]

Durum/sınıf dengesizliği problemini çözmek üzere birçok yaklaşım önerilmiş olmakla birlikte, bütün olası problemler ve dengesiz veri kümeleri için üzerinde anlaşılan genel bir çerçeve yoktur. Aslında, bu problemler için önerilen çözümlerin tümünü bir arada karşılaştırmak ve en üstün yöntemin hangisi olduğunu ortaya koymak için bir çerçeve oluşturmak da olası değildir. Bunun nedeni, problemi tamamen ortadan kaldırabilmek hedeflenmese de derecesini azaltmak için izlenebilecek yolların çok sayıda etkene bağlı (çok boyutlu) olmasıdır. Örneğin; her problem farklıdır, bazı seyreklik giderme yöntemleri uygulanabilirken bazıları uygulanamaz, aynı problem için öğrenme sürecinde kullanılan verinin farklılaşması sonuçları etkileyebilir, farklı yapay öğrenme yöntemleri aynı çözüm yaklaşımı için farklı davranabilir, bir veri kümesinde üstün olduğu gözlemlenen bir yöntem bir başkası için daha zayıf olabilir, aynı problem için aynı verinin kullanıldığı karşılaştırmalarda bile yöntemler bağlı oldukları parametrelerin değerleri değiştirildiğinde farklı sonuçlar verebilir, problem-tabanlı olan başarım ölçevlerinin ne olduğu seçilecek yöntemin hangisi olacağı konusunda belirleyici olabilir, vs. Seyreklik işleme yöntemlerinin başarımlarının karşılaştırılmasının sayılan etkenlerin ve belki daha fazlasının eşitlendiği koşullarda yapılması, sağlıklı ve tutarlı sonuçlar için doğru olacaktır. Bununla ilgili bir örnek, destek-vektör makineleri sınıfandırıcısı ile fetal kafataslarından spina bifida patolojisini belirleme [17] problemiyle ilgili olarak Tablo 4'te verilmektedir. Bu örnekte, ilk öğrenme kümesi aynı kalmak koşuluyla, borderline-SMOTE [42] ve rasgele altörnekleme birlikte kullanılarak, farklı örnekleme parametre değerleri için ve değişen rasgele altörnekleme yüzdeleri gözetilerek öğrenme ve sınama verisinde görülen AUC [14] değerleri sunulmaktadır.

Tablo 4: Bütün borderline-SMOTE örneklemeleri için AUC

VERİ / Oran	*%0*	*%100*	*%200*	*%300*	*%400*	*%500*
Öğrenme	0.619	0.754	0.793	0.815	0.823	0.835
Sınama	0.787	0.837	0.882	0.885	0.896	0.896

Bu makalede değinilen bütün yaklaşımların ve yöntemlerin farklı yönlerden üstünlük ve zayıflık gösteren yanları bulunmaktadır. Aşağıdakiler bu durumun örnekleridir:

- Gerçeklemesi kolay olan örnekleme yöntemleri sınıflandırıcıya bağlı değildir, ama altörneklemede değerli veri kaybedilebilir, üstörneklemede ise hem zaman maliyeti hem de üstuydurma söz konusudur.
- İteleme yöntemleri, çoğu zaman üstuydurmadan kaçınmayı sağlar ama gürültülü veri ve aykırı değerlere duyarlılığı istenmeyen bir özelliktir.
- Maliyete-duyarlı öğrenmede, yanlılık tercihi seyrek sınıf yönünde yapılarak yanlış sınıflandırma maliyeti en aza çekilebilir, ama gerçek hata maliyetlerini bilmek veya bulmak genellikle zordur.
- Kolektif sınıflandırıcılar, tek bir sınıflandırıcıdan daha iyi başarım sağlar ve gürültüye dirençlidir, ama zaman ve üstuydurma yönünden zayıflık gösterir.
- Tanıma yöntemleri, çok öznitelik barındıran veri için uygun olmakla beraber, birçok sınıflandırıcı 1-sınıf sınıflandırmada kullanılamamaktadır.

Sonuç olarak, seyrek öğrenme kümeleri ile öğrenme gerçekleştirmesi gereken veri madenciliği problemlerinde, sınıflandırıcı tasarımında birçok etkenin gözetilmesi ve uygun yöntemlerle sürecin ele alınması gerekir. Çözümler, genellikle probleme dayalı olup, bir alanda iyi sonuç veren seçeneklerin başka bir alanda aynı başarıyı sağlaması beklenmez. Bir sınıflandırıcıyı tasarlamak üzere yola çıkanların, başarım ölçevlerinin ne anlama geldiği ve nasıl kullanılması gerektiğinden başlayarak, eldeki problemi iyi özümsemeleri, özel durumların farkında olarak seyrekliği gidermek için gerekli olan uygun yöntemleri uygulamaları ve tasarımı gerçeklemeleri gerekir.

6. Kaynaklar

[1] M. Kubat, R.C. Holte ve S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images", *Machine Learning*, 30(2-3), 195-215, 1998.

[2] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys*, 34(1), 1-47, 2002.

[3] P. Lynch, "The Origins of Computer Weather Prediction and Climate Modeling", *Journal of Computational Physics*, 227(7), 3431-3444, 2008.

[4] A. Kumar, "Computer Vision-based Fabric Defect Detection: A Survey", *IEEE Transactions on Industrial Electronics*, 55(1), 348-363, 2008.

[5] A. Patel, Q. Qassim ve C. Wills, "A Survey of Intrusion Detection and Prevention Systems", *Information Management & Computer Security*, 18(4), 277-290, 2010.

[6] M.A. Farajian ve S. Mohammadi, "Mining the Banking Customer Behavior using Clustering and Association Rules Methods", *International Journal of Industrial Engineering & Production Research*, 21(4), 239-245, 2010.

[7] R. Takahashi, ve Y. Kajikawa, "Computer-aided Diagnosis: A Survey with Bibliometric Analysis", *International Journal of Medical Informatics*, 101, 58-67, 2017.

[8] T. Fawcett, "An Introduction to ROC Analysis", *Pattern Recognition Letters*, 27, 861-874, 2006.

[9] J.A. Swets, "Measuring the Accuracy of Diagnostic Systems", *Science*, 240(4857), 1285-1293, 1988.

[10] J. Han, Kamber, M. ve Pei, J., Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, Waltham, Massachusetts, 2012.

[11] E. Alpaydın, Introduction to Machine Learning, The MIT Press, Cambridge, Massachusetts, 2014.

[12] C.J. van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.

[13] I. Kononenko ve I. Bratko, "Information-based Evaluation Criterion for Classifier's Performance", *Machine Learning*, 6, 67-80, 1991.

[14] A. Bradley, "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms", *Pattern Recognition*, 30(7), 1145-1159, 1997.

[15] G.M. Weiss, "Mining with Rarity: A Unifying Framework", *ACM SIGKDD Explorations Newsletter*, 6(1), 7-19, 2014.

[16] U. Konur, F. S. Gürgen, F. Varol ve L. Akarun, "Computer Aided Detection of Spina Bifida using Nearest Neighbor Classification with Curvature Scale Space Features of Fetal Skulls Extracted from Ultrasound Images", *Knowledge Based Systems*, 85, 80-95, 2015.

[17] U. Konur, "Computerized Detection of Spina Bifida using SVM with Zernike Moments of Fetal Skulls in Ultrasound Screening", *Biomedical Signal Processing and Control*, 43, 18-30, 2018.

[18] R. Longadge, S.S. Dongre ve M. Malik, "Class Imbalance Problem in Data Mining: Review", *International Journal of Computer Science and Network*, 2(1), 2013.

[19] S. Maheshwari, R.C. Jain ve R.S. Jadon, "A Review on Class Imbalance Problem: Analysis and Potential Solutions", International Journal of Computer Science Issues, 14(6), 43-51, 2017.

[20] S.S. Dongre ve L.G. Malik, "Rare Class Problem in Data Mining: Review", *International Journal of Advanced Research in Computer Science*, 8(7), 1102-1105, 2017.

[21] M.V. Joshi, "On Evaluating Performance of Classifiers for Rare Classes", *IEEE International Conference on Data Mining*, 641-644, 2002.

[22] G.M., Weiss, "Timeweaver: A Genetic Algorithm for Identifying Predictive Patterns in Sequences of Events", *Annual Conference on Genetic and Evolutionary Computation* 718-725, 1999.

[23] D.R. Carvalho ve A.A. Freitas, "A Genetic Algorithm for Discovering Small-disjunct Rules in Data Mining", *Applied Soft Computing*, 2(2), 75-88, 2002.

[24] N. Japkowicz, ve S. Stephen, "The Class Imbalance Problem: A Systematic Study", *Intelligent Data Analysis*, 6(5), 429-449, 2002.

[25] Estabrooks, A. ve N. Japkowicz, "A Mixture-of-experts Framework for Learning from Imbalanced Data Sets", *International Symposium on Intelligent Data Analysis*, 34-43, 2001.

[26] M.V. Joshi, R.C. Agarwal ve V. Kumar, "Mining Needles in a Haystack: Classifying Rare Classes via Two-phase Rule Induction", *ACM SIGMOD Conference on Management of Data*, 91-102, 2001.

[27] M.V. Joshi, R.C. Agarwal ve V. Kumar, "Predicting Rare Classes: Can Boosting Make any Weak Learner Strong?", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 297-306, 2002.

[28] J.R. Quinlan, "Improved Estimates for the Accuracy of Small Disjuncts", *Machine Learning*, 6, 93-98, 1991.

[29] M. di Martino, A. Fernãndez, P. Iturrralde ve F. Lecumberry, "Novel Classifier Scheme for Imbalanced Problems", *Pattern Recognition Letters*, 34(10), 1146-1151, 2013.

[30] R.C. Holte, L.E. Acker ve B.W. Porter "Concept Learning and the Problem of Small Disjuncts", *International Joint Conference on Artificial Intelligence*, 813-818, 1989.

[31] K.M. Ting, "The Problem of Small Disjuncts: Its Remedy in Decision Trees", *Canadian Conference on Artificial Intelligence*, 91-97, 1994.

[32] A. van den Bosch, Weijters, T., van den Herik, H.J. ve Daelemans, W., "When Small Disjuncts Abound, Try Lazy Learning: A Case Study", *Belgian-Dutch Conference on Machine Learning*, 109-118, 1997.

[33] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing, Boston, Massachusetts, 1989.

[34] P.A.A. Resende ve A.C. Drummond, "Adaptive Anomaly-based Intrusion Detection System using Genetic Algorithm and Profiling", *Security and Privacy*, 1(4), 1-13, 2018.

[35] P. Riddle, R. Segal ve O. Etzioni, "Representation Design and Brute-force Induction in a Boeing Manufacturing Design", *Applied Artificial Intelligence*, 8, 125-147, 1994.

[36] R. Agrawal, T. Imielinski ve A. Swami, "Mining Association Rules between Sets of Items in Large Databases", *ACM SIGMOD International Conference on Management of Data*, 207-217, 1993.

[37] Y. Sun, M..S. Kamel ve Y. Wang, "Boosting for Learning Multiple Classes with Imbalanced Class Distribution", *International Conference on Data Mining*, 592-602, 2006.

[38] J.H. Zhao, X. Li ve Z.Y. Dong, "Online Rare Events Detection", *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1114-1121, 2007.

[39] F.D. Frumosu, A.R. Khan, H. Schiøler, Külahçı, M., Zaki, M. ve Westermann-Rasmussen, P., "Cost-sensitive Learning Classification Strategy for Predicting Product Failures", *Expert Systems with Applications*, 161, 2020.

[40] M. Kubat ve S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-sided Selection", *International Conference on Machine Learning*, 1997.

[41] N.V. Chawla, K.W. Bowyer, L.O. Hall ve W.P. Kegelmeyer "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, 16, 321-357, 2002.

[42] H. Han, W.Y. Wang ve B.H. Mao, "Borderline-SMOTE: A New Over-sampling Method in Imbalanced Data Sets Learning", *International Conference on Advances in Intelligent Computing*, 878-887, 2005.

[43] S. García, J. Derrac, I. Triguero, C.J. Carmona ve F. Herrera, "Evolutionary-based Selection of Generalized Instances for Imbalanced Classification", *Knowledge Based Systems*, 25(1), 3-12, 2012.

[44] B. Das, N.C Krishnan ve D.J. Cook, "RACOG and wRACOG: Two Probabilistic Oversampling Techniques", *IEEE Transactions on Knowledge and Data Engineering*. 27(1), 222-234, 2015.

[45] G. Casella ve E.I. George, "Explaining the Gibbs Sampler", *The American Statistician*, 46(3), 167-174, 1992.

[46] S. Susan ve A. Kumar "The Balancing Trick: Optimized Sampling of Imbalanced Datastes – A Brief Survey of the Recent State of the Art", *Engineering Reports*, 2020.

[47] R.E. Schapire "A Brief Introduction to Boosting", *International Joint Conference on Artificial Intelligence*, 1-6, 1999.

[48] M.V. Joshi, V. Kumar ve R.C. Agarwal, "Evaluating Boosting Algorithms to Classify Rare Cases: Comparisons and Improvements", *IEEE International Conference on Data Mining*, 257-264, 2001.

[49] N.V. Chawla, A. Lazarevic, L.O. Hall ve K.W. Bowyer, "SMOTEBoost: Improving Predition of the Minority Class in Boosting", *European Conference on Principles and Practice of Knowledge Discovery in Databases*, 107-119, 2003.

[50] W. Fan, S.J. Stolfo, J. Zhang, ve P.K. Chan, "AdaCost: Misclassification Cost-sensitive Boosting", *International Conference on Machine Learning*, 97-105, 1999.

[51] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi ve M. Asadpour, "Boosting Methods for Multi-class Imbalanced Data Classification: An Experimental Review", *Journal of Big Data*, 7(70), 2020.

[52] L. Breiman, "Bagging Predictors", *Machine Learning*, 24, 123-140, 1996.

[53] H. Guo ve H.L. Viktor, "Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach", *ACM SIGKDD Explorations Newsletter*, 6(1), 30-39, 2004.

[54] P. Kang ve S. Cho, "EUS SVMs: Ensemble of Under-sampled SVMs for Data Imbalance Problems", *International Conference on Neural Information Processing*, 837-846, 2006.

[55] X.Y. Liu, J. Wu ve Z.H. Zhou "Exploratory Undersampling for Class-imbalance Learning", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics,)* 39(2), 539-550, 2009.

[56] H. Guo, J. Zhou ve C-a. Wu, "Ensemble Learning via Constraint Projection and Undersampling Technique for Class-imbalance Problem", *Soft Computing*, 24, 4711-4727, 2019.

[57] L. Lusa ve R. Blagus, "The Class-imbalance for High-dimensional Class Prediction", *International Conference on Machine Learning and Application*, 123-126, 2012.

[58] K. Chomboon, K. Kerdprasop ve N. Kerdprasop, "Rare Class Discovery Techniques for Highly Imbalanced Data", *International MultiConference of Engineers and Computer Scientists*, 2013.

[59] D. Mladenic ve M. Grobelnik, "Feature Selection for Unbalanced Class Distribution and Naive Bayes", *International Conference on Machine Learning*, 258-267, 1999.

[60] M. Wasikowski ve X-w. Chen, "Combating the Small Sample Class Imbalance Problem using Feature Selection", *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1388-1400, 2010.

[61] Y. Yang ve J.O. Pedersen, "A Comparative Study on Feature Selection for Text Categorization", *International Conference on Machine Learning*, 412-420, 1997.

[62] X. Jin, A.Xu, R. Bie, ve P. Guo, "Machine Learning Techniques and Chi-Square Feature Selection for Cancer Classification using SAGE Gene Expression Profiles", *International Workshop on Data Mining for Biomedical Applications*, 106-115, 2006.

[63] M.F. Caropreso, S. Matwin ve F. Sebastiani, "A Learner-independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization", *Text Databases and Document Management: Theory and Practice (ed: Chin, A.G.)*, Idea Group Publishing, Hershey, Pennsylvania, 78-102, 2001.

[64] C. Shang, M. Li, S. Feng, Q. Jiang ve J. Fan, "Feature Selection via Maximizing Global Information Gain for Text Classification", *Knowledge Based Systems*, 54, 298-309, 2013.

[65] K. Kira ve L.A. Rendell, "The Feature Selection Problem: Traditional Methods and New Algorithms", *AAAI Conference on Artificial Intelligence*, 129-134, 1992.

[66] I. Kononenko, "Estimating Attributes: Analysis and Extension of RELIEF", European Conference on Machine Learning, 171-182, 1994.

[67] Z. Zheng, X. Wu ve R. Srihari,"Feature Selection for Text Categorization on Imbalanced Data", *ACM SIGKDD Explorations Newsletter*, 6(2), 80-89, 2004.

[68] Ş. Ertekin, J. Huang ve C.L. Gilles, "Active Learning for Class Imbalance Problem", *ACM SIGIR Conference on Research and Development in Information Retrieval*, 823-824, 2007.

[69] X-w. Chen ve M. Wasikowski, "FAST: A Roc-based Feature Selection Metric for Small Samples and Imbalanced Data Classification Problems", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 124-132, 2008.

[70] M. Alibeigi, S. Hashemi ve A. Hamzeh, "DBFS: An Effective Density Based Feature Selection Scheme for Small Sample Size and High Dimensional Imbalanced Data Sets", *Data and Knowledge Engineering*, 81-82(1), 67-103, 2012.

[71] L. Yin, Y. Ge, K. Xiao, X. Wang ve X. Quan, "Feature Selection for High Dimensional Imbalanced Data", *Neurocomputing*, 105, 3-11, 2013.

[72] M.S. Nikulin, "Hellinger Distance", *Encyclopedia of Mathematics*, EMS press, 2001.

[73] A. Jovic, K. Brkic ve N. Bogunovic, "A Review of Feature Selection Methods with Applications", *International Convention on Information and Communication Technology, Electronics and Microelectronics*, 1200-1205, 2015.

[74] G. Dong ve J. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 43-52, 1999.

[75] H. Alhammady ve K. Ramamohanarao, "The Application of Emerging Patterns for Improving the Quality of Rare-class Classification", *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 207-211, 2004.

[76] H. Alhammady ve K. Ramamohanarao, "Using Emerging Patterns and Decision Trees in Rare-class Classification", *IEEE International Conference on Data Mining*, 315-318, 2004.

[77] H. Alhammady, "A Novel Approach for Mining Emerging Patterns in Rare-class Datasets", *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering (ed: Sobh, T.)*, 207-211, 2007.

[78] A. Fernãndez, M.J. del Jesus ve F. Herrera, "On the Influence of an Adaptive Inference System in Fuzzy Rule Based Classification Systems for Imbalanced Data-sets", *Expert Systems with Applications*, 36(6), 9805-9812, 2009.

[79] N. Japkowicz, C. Myers ve M. Gluck, "A Novelty Detection Approach to Classification", *International Joint Conference on Artificial Intelligence*, 518-523, 1995.

[80] B. Raskutti ve A. Kowalczyk, "Extreme Re-balancing for SVMs: A Case Study", *ACM SIGKDD Explorations Newsletter*, 6(1), 60-69, 2004.

[81] W.W. Cohen, "Fast Effective Rule Induction", *International Conference on Machine Learning*, 115-123, 1995.

[82] N. Japkowicz, "Learning from Imbalanced Data Sets: A Comparison of Various Strategies", *AAAI Workshop on Learning from Imbalanced Data Sets*, 10-15, 2000.

[83] H. Lee ve S. Cho, "The Novelty Detection Approach for Different Degrees of Class Imbalance", *International Conference on Neural Information Processing*, 21-30, 2006.

[84] C. Bellinger, S. Sharma ve N. Japkowicz, "One-class versus Binary Classification: Which and When?", *International Conference on Machine Learning and Applications*, 102-106, 2012.

[85] N. Japkowicz, "Supervised Learning with Unsupervised Output Separation", *International Conference on Artificial Intelligence and Soft Computing*, 321-325, 2002.

[86] J. Wu, H. Xiong, P. Wu ve J. Chen, "Local Decomposition for Rare Class Analysis", *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 814-823, 2007.

[87] R. Gong ve S.R. Huang, "A Kolmogorov-Smirnov Statistic based Segmentation Approach to Learning from Imbalanced Datasets: With Application in Property Refinance Prediction", *Expert Systems with Applications*, 39(6), 6192-6200, 2012.

[88] R. Agrawal ve R. Srikant, "Fast Algorithms for Mining Association Rules", *International Conference on Very Large Databases*, 487-499, 1994.

[89] B. Liu, W. Hsu ve Y. Ma, "Mining Association Rules with Multiple Minimum Supports", ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining*, 337-341, 1999.

[90] Y.S. Koh ve N. Rountree, "Finding Sporadic Rules using Apriori-inverse", *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 97-106, 2005.

[91] L. Szathmary, A. Napoli ve P. Valtchev, "Towards Rare Itemset Mining", *International Conference on Tools with Artificial Intelligence*, 305-312, 2007.

[92] R. Batuwita ve V. Palade, "FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning", *IEEE Transactions on Fuzzy Systems*, 18(3), 558-571, 2010.

[93] R. Kohavi, "Data Mining with MineSet: What Worked, What did not Work, and What might", *International Conference on Knowledge Discovery and Data Mining*, 1-6, 1998.

[94] I. Kopanas, N.M. Avouris ve S. Daskalaki, "The Role of Domain Knowledge in a Large Scale Data Mining Project", *Hellenic Conference on AI: Methods and Applications of Articial Intelligence*, 288-299, 2002.

[95] M.D. Peréz-Godoy, F. Alberto, A.J. Rivera ve M.J. del Jesus, "Analysis of an Evolutionary RBFN Design Algorithm, CO^2RBFN, for Imbalanced Data Sets", *Pattern Recognition Letters*, 31(15), 2375-2388, 2010.

[96] J. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, California, 1993.

[97] Y.Y. Nguwi ve S.Y. Cho, "An Unsupervised Self-organizing Learning with Support Vector Ranking for

Imbalanced Datasets", *Expert Systems with Applications*, 37(12), 8303-8312, 2010.
[98] A. Ultsch ve F. Mörchen, "ESOM-maps: Tools for Clustering, Visualization and Classification with Emergent SOM", *Teknik rapor 46,* Matematik ve Bilgisayar Bilimi Bölümü, Marburg Üniversitesi, 2005.
[99] T.S.F. Haines ve T. Xiang, "Active Rare Class Discovery and Classification using Dirichlet Processes", *International Journal of Computer Vision*, 106, 315-331, 2014.
[100] K.K. Wankhade, K.C. Jondhale ve V.R. Thool, "A Hybrid Approach for Classification of Rare Class Data", *Knowledge and Information Systems*, 56, 197-221, 2017.
[101] M. Koziarski, M. Wozniak ve B. Krawczyk, "Combined Cleaning and Resampling Algorithm for Multi-class Imbalanced Data with Label Noise", *Knowledge Based Systems*, 204, 2020.
[102] J.M. Johnson ve T.M. Koshgoftaar, "Survey on Deep Learning with Class Imbalance", *Journal of Big Data*, 6(27), 2019.
[103] J.L. Leevy, T.M. Khoshgoftaar, R.A. Bauder ve N. Seliya, "A Survey on Addressing High Class Imbalance in Big Data", *Journal of Big Data*, 5(42), 2018.
[104] J.M. Johnson ve T.M. Koshgoftaar, "Medicare Fraud Detecrion using Neural Networks", *Journal of Big Data*, 6(1), 2019.
[105] A.C. Schierz, "Virtual Screening of Bioassay Data", *Jounal of Cheminformatics*, 1(21), 2009.

Dr. Umut KONUR

Umut Konur; lisans derecesini (BSc.) 2003'te, yüksek lisans derecesini (MSc.) 2006'da ve doktora derecesini (PhD) 2015'te Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü'nden almıştır. 2018'den bu yana Zonguldak Bülent Ecevit Üniversitesi Bilgisayar Mühendisliği Bölümü'nde öğretim üyesi olarak çalışmaktadır. Araştırma ve ilgi alanları arasında veri madenciliği, görüntü işleme, bilgisayarla görme, bilgisayar destekli tanı ve karar destek sistemleri bulunmaktadır.