SMOTE Algoritması ve SVM Sınıflandırması

İle Veri Madenciliği Uygulaması

Muhammet Yasin ELMAS, Hilal BORAZAN

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

**Anahtar Kelimeler─** SMOTE algoritması, SVM sınıflandırması

**1.GİRİŞ (INTRODUCTION)**

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

Literatürde bu çalışmadakine benzer düzeyde çalışmalar yapılmıştır. Berzal ve arkadaşlarının geliştirmiş oldukları Tminer yazılımı bir framework olup daha çok veri tabanı sistemleri üzerinde çalışmaktadır. Nesne ve model oluşturabilme özelliği bulunmaktadır [1]. Fdez ve arkadaşlarının geliştirmiş oldukları KEEL isimli yazılım veri madenciliği ve bulanık sistemler için bir araç niteliği taşımaktadır. Fdez çalışmasında birçok avantajından bahsetmiştir. Veri yönetimine tam anlamıyla izin veren ve her türlü veri seti ile çalışabilen bir yazılım geliştirmişlerdir [2]. Gorodetsky ve arkadaşları danışman tabanlı dağıtık veri madenciliği alanında kullanılabilecek bir yazılım aracı geliştirmişlerdir. Meta bilgiler ve kaynak tabanlı olarak çalışmaktadır. Gorodetsky’nin geliştirdiği yazılımın sınırlayıcı yanı sadece dağıtık (distributed) veri kaynakları üzerine çalışabiliyor oluşudur [3]. Bose ve arkadaşının geliştirmiş oldukları IDM ortamı, bu ortam danışman tabanlı veri madenciliği ortamı olup kullanıcı ara yüzü geliştirmişlerdir. Tasarladıkları kayıt ve mesaj panelleri ile danışmanların (agent) birbirleri ile haberleşmesi sağlanmıştır. Önerdikleri sistem bir karar destek sistemidir ve çalışmalarında bu sistemin avantaj ve limitlerinden bahsetmişlerdir. Limite bir örnek olarak kullanıcı tabanlı bir sistem olduğu için kullanıcının doğru amaçları girmesi beklenmektedir [4]. Java’da yazılmış bir başka veri madenciliği yazılımı Bhat ve arkadaşları tarafından geliştirilmiş olup veri seti olarak sadece twitter üzerinde çalışmaktadır. Bhat ve arkadaşlarının yaptıkları çalışmada Dünya üzerinde seçilen bir bölge için gönderilen tweet dağılımını çıkarabilmektedirler [5]. Robu ve arkadaşları Weka yazılımı için bir dönüştürücü araç geliştirmişlerdir. Weka açık kaynak kodlu bir yazılım olduğundan araç geliştirip ekleme yapmak kolay olmatadır. Robu ve arkadaşlarının geliştirdikleri araç, Weka’nın okuyamayacağı veri tabanlarından Weka’nın okuyabileceği veri tabanlarına dönüşüm yaparak kullanıcının işini kolaylaştırmaktadır. MySql, MsSql gibi çeşitli veri tabanlarından okuma yapıp sağladığı ara yüzler ile veri üstünde istenen filtreler uygulanıp Weka’nın kullanabileceği veri seti tarzına dönüşüm yapılmaktadır [6].

Bu çalışmada geliştirilen yazılım sayesinde kullanıcı elindeki veri setini programa okutup SMOTE algoritmasını uygulayarak SVM sınıflandırmasındaki değişimi gösteren bir yazılım geliştirilmiştir.

**2. VERİ MADENCİLİĞİ (DATA MINING)**

Veri madenciliği sayesinde elde bulunan veri üzerinde çeşitli algoritmalar uygulayıp veri hakkında farklı kararlara ulaşılabilir. Veri madenciliği çoğunlukla istatistiksel hesaplamalara dayanır. Sınıflama, kümeleme ve karar ağaçları oluşturma gibi çeşitli yöntemlere sahiptir. Veri madenciliği gözlemsel veri kümelerinin genelde büyük hacimli olanların analizini yapıp beklenmedik ilişkileri bulmak, yeni yöntemler kullanarak anlaşılabilir ve faydalı bir biçime sokmak olarak tanımlanabilir [6]. İki önemli veri analiz metodu olan sınıflandırma ve karar verme, önemli veri sınıflarını belirleyebilir veya yapıcı modeller oluşturabilirler. Böylelikle gelecekteki veri hakkında tahmin ve yorum yapılabilmesini sağlayabileceklerdir. Sınıflandırma kategorik verilerin tahmininde kullanılır. Sınıflandırmada genellikle karar ağaçları, yapay sinir ağları ve genetik algoritmalar gibi teknikler kullanılır [7,8,9].

**3. Metod**

1. **SMOTE algoritması**
2. **SVM sınıflandırması**
3. **WEKA veri madenciliği uygulama geliştirme platformu**
4. **Veri Kümesi**

**3.1.SMOTE (Synthetic Minority Over-sampling Technique)**

SMOTE algoritması ile veri kümeleri sentetik veri artırım daha dengeli hale getirilir.

Verilen bir sınıfın gerçek bir nesnesi ve en yakın komşularından biri (aynı sınıftan) arasında sentetik satırlar oluşturur. Daha sonra bu iki nesne arasındaki çizgi boyunca bir nokta seçer ve rastgele seçilen bu noktaya dayanarak yeni nesnenin niteliklerini (hücre değerleri) belirler.

**SMOTE Algoritması Sözde Kodu(Pseudo-Code)**

**Algorithm** SMOTE(T,N,k)

**Input**: T; Azınlık sınıfı örneklem sayısı

N%; Smote miktarı

k; komşu sayısı

**Output**: (N/100)\*T //sentetik azınlık sınıfı örnekleri

(N, %100'den az ise, azınlık sınıfı örneklerini rastgele seçmelisiniz)

**if** N<100

**then** T azınlık sınıfı örneklerini randomize etmek

T=(N/100)\*T

N=100

**endif**

**N**=(int)(N/100) //SMOTE miktarının 100'ün ayrılmaz katlarında oldusğu varsayılmaktadır.

**k**=komşu sayısı

**nummattrs**=özellik(attribute) sayısı

**Sample[][]**:Orijinal azınlık sınıfı örnekleri için dizi

**Synthetic[][]:**Sentetik örnekler için dizi

**for** i<-1 **to** T

i için k komşu değerlerini hesapla ve dizinin indislerine kaydet

**Populate**(N,i,narray) //Fonksiyon sentetik değerler üretir

**endfor**

**while** N=!0

1 ile k arasında rastgele bir sayı seç, nn değerini çağır.

Bu adım i'nin en yakın komşularından birini seçer.

**for** attr<-1 to numattrs

**hesapla: dif**=Sample[nnarray[nn][attr]-Sample[i][0]

**hesapla: gap**=0 ile 1 arasında rasgele sayı

Synthetic[newindex][attr]=Sample[i][attr]+gap\*dif

**endfor**

newindex++

N=N-1

**endwhile**

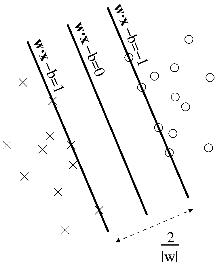
**return**(end of populate)

**End of Pseudo-Code**

**3.2. SVM(SUPPORT VEKTÖR MACHINE)**

Sınıflandırma (Classification) konusunda kullanılan oldukça etkili ve basit yöntemlerden birisidir. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır. İşte SVM bu sınırın nasıl çizileceğini belirler.

Bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu sınır çizgileri birbirine yaklaştırılarak ortak sınır çizgisi üretilir.



Bu şekilde iki grup iki boyutlu bir düzlem üzerinde gösterilmiştir. Bu düzlemi ve boyutları birer özellik olarak düşünmek mümkündür. Yani basit anlamda sisteme giren her girdinin (input) bir özellik çıkarımı (feature extraction) yapılmış ve sonuçta bu iki boyutlu düzlemde her girdiyi gösteren farklı bir nokta elde edilmiştir. Bu noktaların sınıflandırılması demek, çıkarılmış olan özelliklere göre girdilerin sınıflanması demektir.

Yukarıda her iki sınıf arasında oluşan aralığa tolerans (offset) demek mümkündür. Bu düzlemdeki her bir noktanın tanımı aşağıdaki gösterim ile yapılabilir:

http://www.bilgisayarkavramlari.com/wp-content/uploads/2008/12/svm_denklem.jpg

Yukarıdaki gösterimi şu şekilde okumak mümkündür. Her x,c ikilisi için X vektör uzayımızdaki bir nokta ve c ise bu noktanın -1 veya +1 olduğunu gösteren değeridir. Bu noktalar kümesi i= 1 ‘den n’e kadar gitmektedir.

Yani bu gösterim bir önceki şekilde olan noktaları ifade etmektedir.

Bu gösterimin bir aşırı düzlem (hyperplane) üzerinde olduğunu düşünürsek. Bu gösterimdeki her noktanın :

wx – b = 0

denklemi ile ifade edilmesi mümkündür. Buradaki w aşırıdüzleme dik olan normal vektörü ve x noktanın değişen parametresi ve b ise kayma oranıdır. Bu denklemi kalsik ax+b doğru denklemine benzetmek mümkündür.

Yine yukarıdaki denkleme göre b/||w|| değeri bize iki grup arasındaki mesafe farkını verir. Bu mesafe farkına daha önce tolerans (offset) ismini de vermiştik. Bu mesafe farkı denklemine göre mesafeyi en yüksek değere çıkarmak için yukarıdaki ilk şekilde gösterilen 0, -1 ve +1 değerlerine sahip 3 doğruyu veren denklemde 2/||w|| formülü kullanılmıştır. Yani doğrular arası mesafe 2 birim olarak belirlenmiştir.

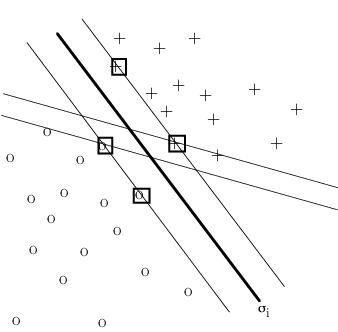
Bu denkleme göre elde edilen iki doğru denklemi:

wx – b = -1

wx + b = 1

olarak bulunmuştur. Aslında bu denklemler doğruların kaydırılması sonucunda elde edilen en yüksek değerlerin bulunması işleminin bir sonucudur. Aynı zamanda bu denklemlerle problemin doğrusal ayrılabilir (linearly seperable) olduğu da kabul edilmiş olur.

Tahmin edileceği üzere iki grup arasındaki aşırıdüzlemin (hyperplane) tek yönlü olması mümkün değildir. Aşağıda bu duruma bir örnek gösterilmiştir:



Yukarıdaki şekilde iki farklı hiperdüzlem (aşırı düzlem) olasılığı bulunmasına karşılık SVM yönteminde bu olasılıklardan en büyük toleransa (offset) sahip olanı alınır.

**3.3. WEKA veri madenciliği uygulama geliştirme platformu**

WEKA bir proje olarak başlayıp bugün dünya üzerinde birçok insan tarafından kullanılmaya başlanan bir Veri Madenciliği uygulaması geliştirme programıdır. WEKA java platformu üzerinde geliştirilmiş açık kodlu bir programdır. WEKA çalıştırıldıktan sonra Şekil 1’de görüldüğü gibi, Application menüsünde çalışılabilecek modlar listelenmektedir. Bunlar komut modunda çalışmayı sağlayan Simple CLI, projeyi adım adım görsel ortamda gerçekleştirmeyi sağlayan Explorer ve projeyi sürükle bırak yöntemiyle gerçekleştirmeyi sağlayan KnowledgeFlow seçenekleridir.

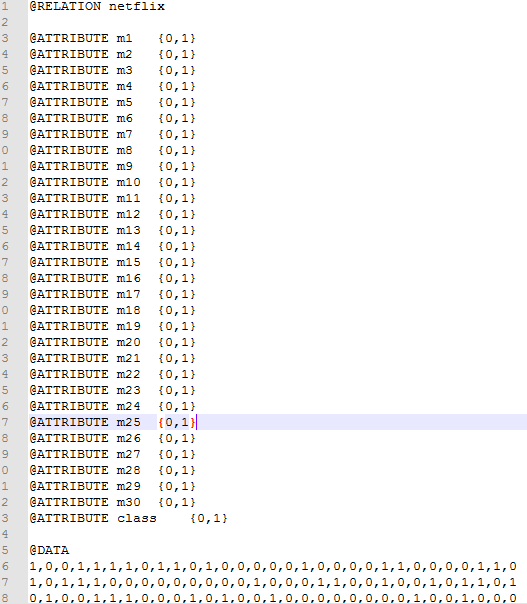


Explorer seçeneği seçildikten sonra üzerinde çalışılacak verilerin seçilmesi, bu veriler üzerinde temizleme ve dönüştürme işlemlerinin gerçekleştirilebilmesini sağlayan ekran ile karşılaşılmaktadır.

Arff, Csv, C4.5 formatında bulunan dosyalar WEKA’da import edilebilir. Herhangi bir text soyadaki verileri WEKA ile işlemek olanaksızdır. Ayrıca Jdbc kullanılarak veritabanına bağlanıp burada da işlemler yapılabilir. WEKA’nın içerisinde Veri İşleme, Veri Sınıflandırma, Veri Kümeleme, Veri İlişkilendirme özellikleri mevcuttur.

Bu adımdan sonra yapılacak olan projenin amacına göre açılan sayfadaki uygun tabdaki (Sınıflandırma, Kümeleme, İlişkilendirme) uygun algoritma veya algoritmalar seçilerek veriler üzerine uygulanmakta ve en doğru sonucu veren algoritma seçilebilmektedir.

**3.4.Veri Kümesi**

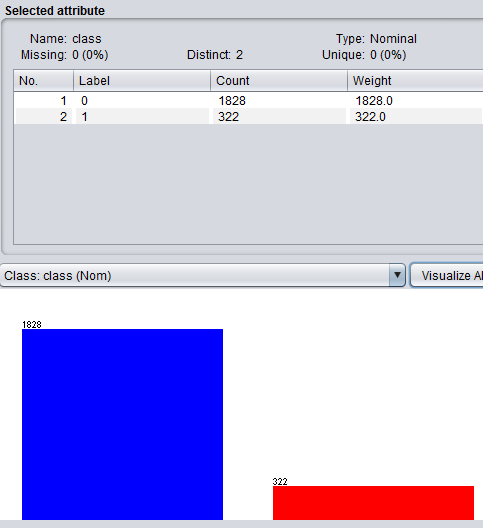


\*M1, m2, m3 ,…,m30 izleyiciler ve filmlere verdikleri tepkiler beğenme-1, beğenmeme-0 .

\*Class yeni eklenen film.

**4.Weka ile Smote algoritması ve SVM sınıflandırması yapılanlar**

*Smote’tan önceki class niteliğinin 0 ve 1 sayıları;*

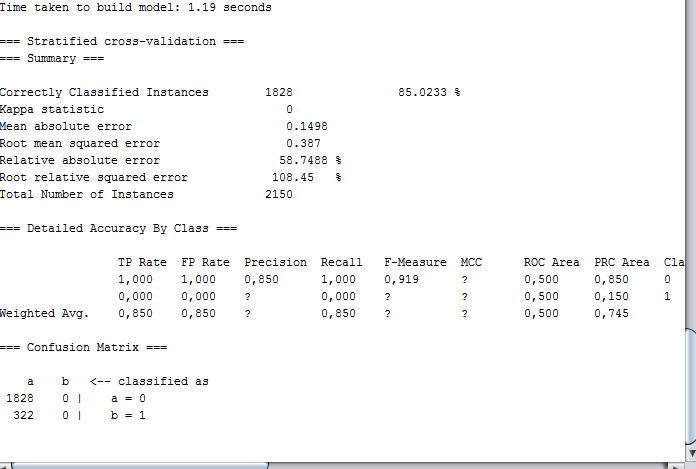


*Smote’tan sonra class niteliğini 0 ve 1 sayıları;*

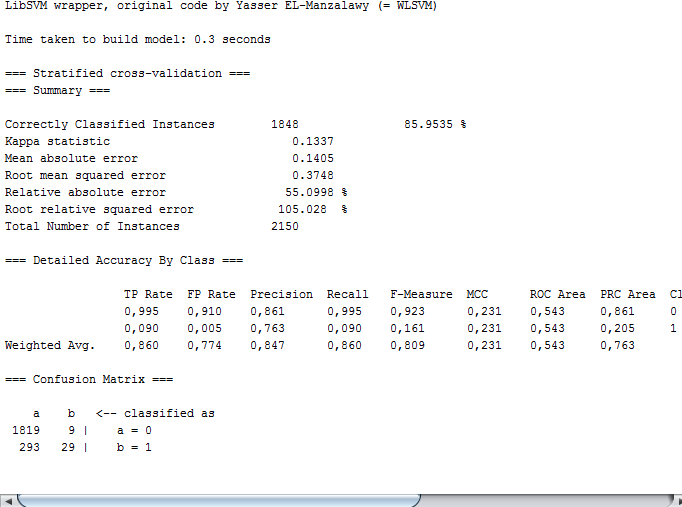


*En düz sınıflandırmalardan olan zeror sınıflandırmasındaki karmaşıklık matrisi ve hata oranları;*

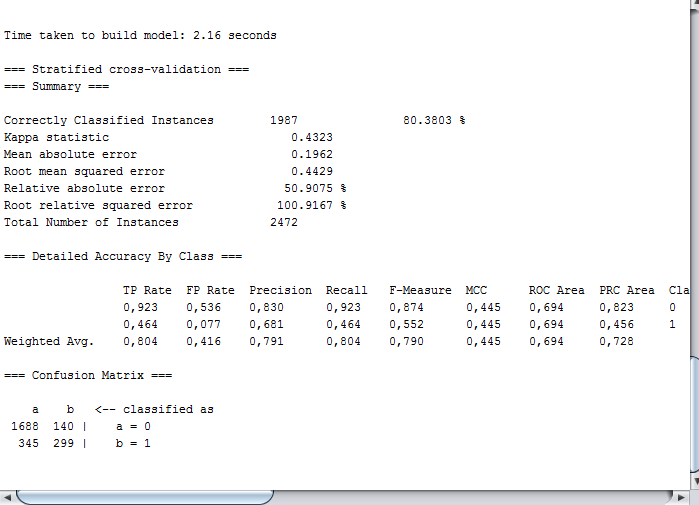
*Smote’tan önce smo uygulayarak alınan karmaşıklık matrisi ve hata oranları;*



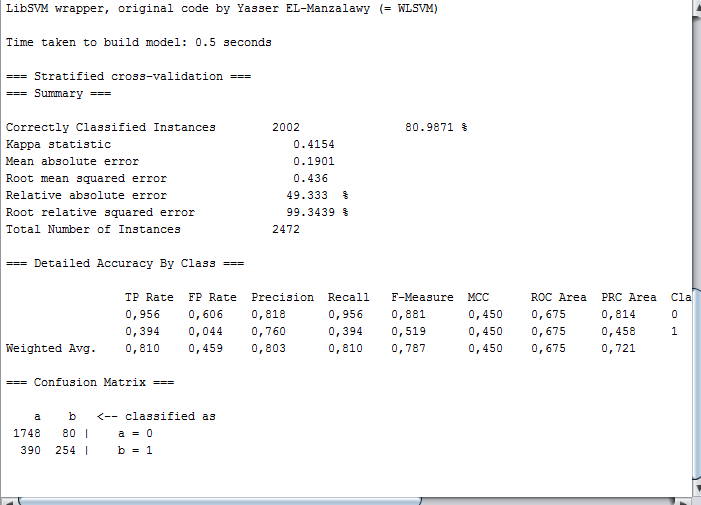
*Smote’tan önce libsvm uygulayarak alınan karmaşıklık matrisi ve hata oranları;*



*Smote’tan sonra smo uygulayarak alınan karmaşıklık matrisi ve hata oranları;*



*Smote’tan sonra libsvm uygulayarak alınan karmaşıklık matrisi ve hata oranları;*



Görsellerde de gördüğümüze göre smote algoritması, smo ve libsvm kullanılarak değişen doğruluk oranları ve karmaşıklık matrislerini görebilmekteyiz.

**5.Sonuçlar(Results)**

-İzleyiciler Netflix’de olan filmleri beğenme yada beğenmeme verilerine bakarak yeni gelen bir filmi beğenip beğenmeyeceğini buluyor.

-Smote algoritması yardımıyla dengesiz veri kümesindeki negatif değerlerin sayısını arttırarak verimizin dengesinin sağlanması gerçekleştirilmiştir.

-Svm sınıflandırılması kullanılarak doğru tahmin etme olasılığı arttırılmıştır.

**Kaynaklar (References)**

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

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

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

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

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

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

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

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

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

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

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