Yeniden Örnekleme Teknikleri Kullanarak SMS Verisi Üzerinde Metin Sınıflandırma Çalışması

Özer Çelik[1], Gürkan Kaplan[2]*,

[1] Eskişehir Osmangazi Üniversitesi Fen-Edebiyat Fakültesi Matematik ve Bilgisayar Bilimleri Bölümü, ESKİŞEHİR

*[2] Eskişehir Osmangazi Üniversitesi Fen-Edebiyat Fakültesi Matematik ve Bilgisayar Bilimleri Bölümü, ESKİŞEHİR

(Alınış / Received: 21.10.2020, Kabul / Accepted: 16.11.2020, Online Yayınlanma / Published Online: 31.12.2020)

Anahtar Kelimeler
Metin Sınıflandırma,
Makine Öğrenmesi,
Yapay Zekâ,
Smote,
SMS

Öz: SMS, mobil cihaz kullanıcılarının iletişimlerinde kullandıkları önemli araçlardan biridir. Günümüzde kullanıcıların almış olduğu çoğu bilginin kaynağı cep telefonlarıdır. Teknolojideki gelişmelerle birlikte cep telefonlarına gelen mesajların içeriği geniş bir alana yayılmakla beraber istenilen kaynaktan gelip gelmediği önemli bir konu teşkil etmektedir. Metin sınıflandırma çalışmalarında Türkçe çalışmaların azlığı dikkat çekicidir. Bu çalışmada çok sayıda kullanıcının telefonlarına gelen mesajlar incelenmiş ve veri ön işleme gibi çeşitli iyileştirme aşamalarından geçirilerek bir araya getirilmiştir. Bu aşamalardan sonra mevcut mesaj içerikleri makine öğrenmesi teknikleri aracılığıyla metin sınıflandırma uygulanarak incelenmiştir. Elde edilen veriler normal, reklam ve spam olacak şekilde 3 farklı kategoriye ayrılmıştır. Ayrıca dengesiz olan veri setini dengeli hale getirmek için Synthetic Minority Oversampling Technique (SMOTE), Condensed Nearest Neighbour (CNN), Undersampling Technique ve Random Undersampling Technique (RUS) uygulanarak sınıflandırma performansları incelenmiştir. 4203 adet SMS'in yer aldığı veri seti üzerinde yapılan çalışma sonucunda en iyi sonucu veren (OACC değerine göre) sınıflandırmalar SMOTE'ta yaklaşık % 80.1 ile Lojistik Regresyon, CNN'de yaklaşık %62.1 ile XGBoost ve RUS'ta yaklaşık %73.8 ile Lojistik Regresyon olmuştur.

Text Classification Study on SMS Data Using Resampling Techniques

Keywords
Text Classification,
Machine learning,
Artificial intelligence,
Smote,
SMS

Abstract: SMS is one of the important tools that mobile devices users use in their communication. Today, most of the information received by users is the source of mobile phones. With the advances in technology, the content of the messages coming to mobile phones is spread over a wide area and whether or not they come from the desired source is an important issue. The lack of Turkish studies in text classification studies is noteworthy. In this study, the messages received from a large number of users' phones were examined and brought together through various improvement stages such as data preprocessing. After that, the existing message contents were examined by applying text classification by machine learning techniques. The data obtained are divided into 3 different categories as normal, advertising and spam. In order to stabilize the unbalanced data set, classification performances were examined by applying Synthetic Minority Oversampling Technique (SMOTE), Condensed Nearest Neighbor (CNN) Undersampling Technique and Random Undersampling Technique (RUS). As a result of the study performed on the data set containing 4203 SMS, the best classifications (according to OACC value) were Logistic Regression with 80.1% in SMOTE, XGBoost with 62.1% in CNN and Logistic Regression with 73.8% in RUS.

*İlgili Yazar, email: gurkan.esogu@gmail.com

1. Giriş

Metin sınıflandırmanın birçok farklı uygulama alanı vardır. Örneğin, kütüphanede yeni bir kitabın konusunun belirlenmesi ve benzer temalı kitaplar arasında uygun yerin belirlenmesi bir metin sınıflandırma problemidir. Bu, insan emeği yerine bilgisayar tarafından yapılırsa, işleme bilgisayarlı metin sınıflandırma denir. Spam filtreleme, bir metnin yazarı veya dilini belirleme, belge indeksleme, kelime anlamını belirleme gibi birçok uygulama metin sınıflandırma uygulamalarına örnektir. Öte yandan, diğer sınıflandırma uygulamaları metin sınıflandırma çözüm yöntemleri ile gerçekleştirilebilir. Konuşmaların sınıflandırılmasının uygulanmasında, konuşma tanıma işleminden sonra metin sınıflandırması yapılarak konuşmanın uygun sınıfa atanması sağlanabilir. Video gibi çoklu ortamların sınıflandırma sorunu, belgedeki çoklu ortamlarla ilgili metinlerin sınıflandırma problemini azaltarak çözülmüştür [1]. Teknolojideki ilerlemelerle birlikte, e-posta, forum ve sohbet odaları gibi bilgisayar destekli iletişim araçlarıyla metin sınıflandırması giderek daha önemli hale geldi. Her ne kadar sürekli güncellenen blog alanları gibi uygulamalar milyonlarca insan tarafından kullanılsa da takibi çok zordur. Geçmişte içerik analizine odaklanan çalışmalar olmasına rağmen, içeriklerin sınıflandırılmasına odaklananların sayısı sınırlıdır. Bunun nedeni, bir metnin havasını sınıflandırmanın zor olmasıdır. Yapay zekâ çalışmalarındaki gelişmeler ile doğal dil işleme kullanılarak bu zorluk ortadan kaldırılmıştır. Doğal dil işleme modelleri, metni sınıflandırmak için önceki bilgilerin kullanılmasını gerektirir. Makine öğrenmesi yaklaşımları, açıklamalı modeller oluşturmak için denetimli öğrenme algoritmaları kullanır. Metin sınıflandırma için, makine öğrenme teknikleri, farklı dillere ve koşullara daha iyi uyum sağlayabileceklerinden, doğal dil işleme tekniklerinden daha iyi sonuçlar elde etme eğilimindedir [2].

Diğer dillerin aksine, literatür araştırıldığında, Türkçe metin sınıflandırma çalışmalarının sayısının az olması dikkat çekicidir. Tüfekçi ve ark., Türk dil bilgisi özelliklerini kullanarak web tabanlı haber metinlerini sınıflandırmak için azaltılmış özellik vektörünü kullandı. Naive Bayes, SVM, C4.5 ve RF sınıflandırma yöntemlerinden elde edilen sonuçlar genellikle daha yüksekti, ancak en yüksek başarı %92.73 ile Naive'den elde edildi [3]. Amasyalı ve diğ. metin sınıflandırma için bir sistem geliştirdi ve %76'lık bir başarı oranı elde etti [4]. Amasya ve Diri, N-gram karakterini kullandı ve metnin yazarını, metnin türünü ve yazarın cinsiyetini belirlemek için bazı sınıflandırma algoritmalarını inceledi. Bu problemlerde başarı sırasıyla %83, %93 ve %96 olarak ölçüldü [5]. Yıldız ve diğ. Metin sınıflandırma için yeni bir özellik çıkarma yöntemi önererek Naive Bayes algoritması ile %96.25 başarı oranı elde etti [6]. Güven ve ark. N-gram kelime belgelerinde Uygulamalı Latent Semantik Analiz yöntemi kullandı [7]. Güran ve diğ. unigram, bigram ve trigram kelimelerinin temsil edildiği ve genellikle yüksek sınıflandırma oranlarına ulaşıldığı bir Türk veri setine çeşitli sınıflandırma yöntemleri uyguladılar. En iyi sonuçlar sırasıyla %95.83, %93.17 ve %52.83 olarak elde edildi [8].

2. Materyal ve Metot

Bu araştırmada, çok sayıda kullanıcının telefonlarına gelen 4203 adet SMS'e (Reklam SMS: 2374, Normal SMS: 1277, Spam SMS: 552) çeşitli makine öğrenme teknikleri ile metin sınıflandırması yapıldı. Bu veri halka açık olarak internet üzerinden Curl yöntemi ile indirilmiştir. SMS'lerin reklam, normal veya spam olmaları ile tahminleme çalışması yapılmıştır. Ayrıca, aşırı örneklemenin ve alt örneklemenin metin sınıflandırma performansına etkisi incelenmiştir. Bu sınıflandırma işleminden önce, veri seti önceden işlenmiş ve gerekli ölçeklendirme yapılmıştır. Bu aşamadan sonra, çeşitli makine öğrenme teknikleri ile SMS'ler için tahmin modelleri kurulmuş ve farklı durumlarda hangi istatistiksel sınıflamanın daha iyi sonuç verdiği tespit edilmiştir. Araştırmada kullanılan veri setine uygulanan makine öğrenme teknikleri için Python(versiyon: 3.7.1) programlama dili kullanılmıştır.

2.1. Makine öğrenmesi teknikleri

Makine öğreniminin amacı, bilgisayarlardaki karmaşık sorunları tespit etmek ve onlara rasyonel çözümler sunmaktır. Bu, makine öğreniminin istatistik, veri madenciliği, örüntü tanıma, yapay zekâ ve teorik bilgisayar bilimi gibi alanlarla yakından ilgili olduğunu ve çok disiplinli bir çalışma gerektirdiğini göstermektedir. Regresyon yöntemleri, bir veya daha fazla açıklayıcı değişken ile bir sonuç değişkeni arasındaki ilişkiyi açıklamak için veri analizinin ayrılmaz bileşenleridir. Makine öğrenimi teknikleriyle yapılan ilk çalışmalardan biri Vapnik tarafından yapılmıştır [9]. Bu çalışmada Vapnik, regresyon problemlerinin çözümü için Destek Vektörü Makinesi kullandı. Bu çalışmada kullanılan yöntem birçok regresyon ve zaman serisi tahmin probleminde yüksek başarı oranları göstermiştir [10]. Günümüzde, teknolojideki gelişmeler ile birlikte makine öğrenme tekniklerinin kullanımı artmaktadır. Aşağıdaki bölüm, bu çalışmada kullanılan makine öğrenme teknikleri ile ilgilidir.

İki veya çok seviyeli kategorik verilerden oluşan çok sayıda sosyo-ekonomik ve tıbbi araştırma sonucunun, bağımlı değişken ile bağımsız değişkenleri arasındaki sebep-sonuç ilişkisini araştırmak için Lojistik Regresyon (LR) Analizi tercih edilir.

K-En Yakın Komşu (KNN) sınıflandırma yönteminde, veriler örnekteki en yakın veriler arasında en sık temsil edilen sınıfa atanır. En yakın veri Öklid uzaklık fonksiyonu hesaplanarak belirlenir. K, bu sınıflandırmada, dikkate alınması gereken en yakın veri sayısıdır [11].

Destek Vektör Makinesi (SVM), çeşitli metin sınıflandırma problemlerinde kullanılan bir makine öğrenme tekniğidir. SVM'ler yapısal risk minimizasyonu ilkesini takip ederler [12].

Naive Bayes (NB), metin üzerinde iyi çalışan basit bir modeldir. Genel olarak sınıflandırma için kullanılan bu yöntem, temel bir olasılık teoremi olan Bayes teoremine dayanmaktadır. Bayes formülü aşağıdaki gibidir.

$$P(A|B) = \frac{P(A|B)P(A)}{P(B)} \tag{1}$$

Bir karar ağacı (DT), bir veri setini ağacın her bir düğümünde tanımlanan bir dizi teste göre tekrar tekrar küçük alt bölümlere ayıran bir sınıflandırma olarak tanımlanır [13].

Rastgele Orman (RF), hem regresyon hem de sınıflandırma görevleri için geniş bir karar ağacı seti kullanan istatistiksel bir öğrenme algoritmasıdır. Yüksek doğruluğu, sağlamlığı ve özelliklerinin sırasına göre bilgi sunma kabiliyeti nedeniyle RF, biyoinformatik ve tıbbi görüntüleme dahil olmak üzere çeşitli makine öğrenme uygulamalarına etkili bir şekilde uygulanır [14].

Adaboost (ADB) sınıflandırmasının temel amacı, bir dizi zayıf öğrenicinin çıktılarını birleştirmektir. Her zayıf öğrenici bir karar ağacı ve yapay bir ağı temsil eder. Her turda, yanlış sınıflandırılmış öğrenicilerin ağırlıkları arttırılır ve doğru sınıflandırılmış öğrenicilerin ağırlıkları azaltılır. Bu işlem, ağırlıklardaki değişiklikler önemsiz hale gelinceye kadar devam eder [15].

Gradyan Artırma (GB) karar ağaçları temelli bir yöntemdir ve önyükleme yığına dayalı rastgele ormandan farklı olan bir gradyan artışına dayanır. Bu yaklaşım genellikle karar ağaçları sabit büyüklükte temel bir öğrenici olarak kullanılır ve bu bağlamda gradyan ağacı güçlendirmesi olarak adlandırılır [16]. Bu makine öğrenme yöntemi, bazı zorlu veri setlerinde en gelişmiş sonuçları sağlamak için yaygın olarak kullanılmaktadır. Bu modelin ana fikri, istenen son kombinasyonu elde etmek için çoklu yineleme yoluyla zayıf sınıflandırıcıları geliştirerek güçlü bir sınıflandırıcı oluşturmaktır. Her yineleme, önceki modelin kalıntılarını azaltır. Kalan yönde yeni bir kombinasyon modeli oluşturarak önceki sonucu iyileştirmek için tasarlanmıştır [17].

Extreme Gradient Boost (XGBoost), daha basit ve daha zayıf bir modelin tahminlerini birleştirerek sonucun tahmin edilmesine yardımcı olan denetimli bir öğrenme algoritmasıdır. Gradyan artırma algoritmasından daha hızlıdır. Daha önce oluşturulmuş zayıf modellerden tekrarlayıp öğrenerek hatayı minimize etmeye çalışır [18].

Yapay sinir ağları (ANN), insan beyninin biyolojik sinir ağlarına dayanarak geliştirilmiştir ve bu ağların fonksiyonlarını yerine getirmek için tasarlanmış bir bilgi işlem sistemidir [19].

2.2. Veri ön işleme

Makine öğrenim teknikleri uygulanmadan önce, aşağıdaki adımlar kullanılarak veri kümesi analiz edilmiştir.

1. Metin Sınıflandırması yapılacağı için eğitime Kullanıcı Adı, Cinsiyet ve PlayStore puanlama bölümleri dahil edilmedi. Yalnızca SMS metinleri alındı.
2. Türk alfabesine göre filtreleme uygulanarak, noktalama işaretleri, sayılar vb. Alfabe dışı karakterler silindi.
3. Her cümleyi bir cümlede belirterek:
    - Tüm büyük harfler küçük harfe dönüştürüldü.
    - Türkçe dilinde duygu belirtmeyen duraklama kelimeleri stopwords filtresi uygulanarak silindi.
    - Her kelimenin kökü TurkishStemmer kullanılarak belirlendi ve cümleler kök formuyla yeniden oluşturuldu.

4. İşlenen matris için en çok kullanılan 1000 kelime CountVectorizer Max Features yöntemi ile belirlendi ve 4203x1000 boyutlu matris oluşturuldu (Reklam SMS: 2374, Normal SMS: 1277, Spam SMS: 552).
5. Veri setindeki SMS metinleri sırasıyla reklam, normal ve spam olacak şekilde üçe bölünmüş ve ilk aşamada dengesiz bir veri seti elde edildi. Bu problemden kurtulmak için veri seti yeniden örneklendi. Dengesiz veri seti, karar ağacı endüksiyon sistemleri veya her sınıfın göreceli dağılımını hesaba katmadan genel doğruluğu optimize etmek için tasarlanmış çok katmanlı sensörler gibi tipik sınıflandırıcılar için zorluklar ortaya çıkarır [20]. Örnekleme yöntemleri, sınıf dengesizliği problemiyle başa çıkmak için yaygın olarak kullanılır. Bu yaklaşımları uygulamak çok kolay olmakla birlikte, onları en etkili şekilde belirlemek birçok zorluğu içerir. Özellikle, aşırı örneklemenin yetersiz örneklemeden daha etkili olup olmadığı ve hangi örnekleme oranının kullanılması gerektiği iyi belirlenmelidir [21].
6. Veri setinde kullanılan test ve eğitim kümeleri sırasıyla 0.3 ve 0.7 olarak belirlendi.
7. SMOTE tekniği ile veri setine aşırı örnekleme (SMOTE) yapıldı.
    - ÖNCE: Reklam SMS: 2374, Normal SMS: 1277, Spam SMS: 552
    - SONRA: Reklam SMS: 2374, Normal SMS: 2374, Spam SMS: 2374
    - CountVectorizer (Max Features parametresi ile 1000 feature seçildi.)
8. Undersampling (CNN) yapıldı.
    - ÖNCE: Reklam SMS: 2374, Normal SMS: 1277, Spam SMS: 552
    - SONRA: Reklam SMS: 757, Normal SMS: 340, Spam SMS: 552
9. Random Undersampling (RUS) yapıldı.
    - ÖNCE: Reklam SMS: 2374, Normal SMS: 1277, Spam SMS: 552
    - SONRA: Reklam SMS: 552, Normal SMS: 552, Spam SMS: 552
10. Üç tekniğin sonuçlarının karşılaştırması yapıldı.

Veri örnekleme algoritmaları için imblearn kütüphanesinin varsayılan parametreleri, ML algoritmaları için scikit learn kütüphanesinin varsayılan parametreleri kullanılmıştır.

Tablo 1. Kullanılan algoritmaların parametreleri

SMOTE	LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2', random_state=0, solver='warn', tol=0.0001, verbose=0, warm_start=False)
CNN	XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bytree=1, gamma=0, learning_rate=0.1, max_delta_step=0, max_depth=3, min_child_weight=1, missing=None, n_estimators=100, n_jobs=1, nthread=None, objective='multi:softprob', random_state=0, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None, silent=True, subsample=1)
RUS	LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='warn', n_jobs=None, penalty='l2', random_state=0, solver='warn', tol=0.0001, verbose=0, warm_start=False)

2.3. İstatistiksel analiz

Araştırmada kullanılan veri setine makine öğrenme teknikleri uygulandıktan sonra doğruluk oranları confusion matrisi kullanılarak hesaplanmıştır. Confusion matrisi, bir veri kümesinde doğru ve yanlış sınıflandırılmış veri gruplarının sayısını veren matristir.

Çalışmamızda çok sınıflı verilerde yaygın olarak kullanılan bir başarı değerlendirme yöntemi olan Genel Doğruluk Oranı (OACC) kullanılmıştır [22].

Tablo 2. Confusion matris

Data Set		Gerçek Durum			Toplam
		Sınıf 1	Sınıf 2	Sınıf 3	
Tahmin	Sınıf 1	a	b	c	j
	Sınıf 2	d	e	f	k
	Sınıf 3	g	h	i	l
Toplam		m	n	o	N

Başarı skorları, confusion matrisi yardımıyla hesaplanmaktadır (Tablo 2). Çalışmamızda kullanılan ve confusion matris yardımıyla hesaplanan başarı ölçüleri ve formülleri (2) numaralı ifadede belirtilmektedir;

$$\begin{aligned} N &= (a + b + c + d + e + f + g + h + i) \\ j &= a + b + c \\ k &= d + e + f \end{aligned} \tag{2}$$

$l = g + h + i$
$m = a + d + g$
$n = b + e + h$
$o = c + f + i$
$OACC = (a + e + i) / N$
$Precision_1 = a / j$
$Precision_2 = e / k$
$Precision_3 = i / l$
$Recall_1 = a / m$
$Recall_2 = e / n$
$Recall_3 = i / o$
$Specificity_1 = (e + f + h + i) / (k + l)$
$Specificity_2 = (a + c + g + i) / (j + l)$
$Specificity_3 = (a + b + d + e) / (j + k)$

confusion matrisi yardımıyla hesaplanan birkaç doğruluk puanı daha vardır. Ayrıca çalışmanın gücü, tip II hata, tip I hata sırasıyla TP değeri, FN değeri ve FP değeri ile hesaplanır.

Tüm analiz ve işlemlerde, Windows 10 64bit işletim sistemine, dört çekirdekli Intel Skylake Core i5-6500 CPU 3.2 GHz 6 MB cahce ve 8 GB 2400 MHz DDR4 Ram belleğe sahip bir bilgisayar kullanılmıştır.

2.4. Örnekleme teknikleri

Dengesizlik, bir sınıfın örneklem büyüklüğünün diğer sınıf veya sınıflardan çok daha yüksek olmasıdır. Bu nedenle, küçük sınıflara ait veri örnekleri ortak sınıflara ait olanlardan daha sık yanlış sınıflandırılmaktadır. Dengesiz veri setini dengelemek için bazı teknikler geliştirilmiştir [23]. Veri analizinde aşırı örnekleme ve örnek alma, bir veri kümesinin sınıf dağılımını ayarlamak için kullanılan tekniklerdir (yani, temsil edilen farklı sınıflar/kategoriler arasındaki oran). Bu terimler hem istatistiksel örneklemede hem anket tasarım metodolojisinde hem de makine öğreniminde kullanılır [24].

2.4.1. Sentetik azınlık örnekleme tekniği (Synthetic Minority Oversampling Technique, Smote)

SMOTE, veri kümenizdeki eleman sayısını dengeli bir şekilde artırmak için kullanılan istatistiksel bir tekniktir. Modül, girdi olarak verdiğiniz mevcut azınlık elemanlarından yeni örnekler oluşturarak çalışır. SMOTE'un bu uygulaması çoğunluk elemanlarının sayısını değiştirmez.
Yeni örnekler sadece mevcut azınlık elemanlarının bir kopyası değil; bunun yerine, algoritma her hedef sınıf ve en yakın komşuları için özellik alanından örnekler alır ve hedef durumun özelliklerini komşularının özellikleriyle birleştiren yeni örnekler oluşturur. Bu yaklaşım, her sınıfa uygun özellikleri arttırır ve örnekleri daha genel yapar. SMOTE, veri kümesinin tamamını girdi olarak alır, ancak yalnızca azınlık elemanlarının sayısını arttırır [25].
Sentetik örnekler, sınıflandırıcının daha küçük ve daha spesifik bölgeler yerine daha büyük ve daha az spesifik karar bölgeleri oluşturmasına neden olur. Azınlık sınıfı örnekleri için, etraflarındaki çoğunluk sınıfı örnekleri tarafından kabul edilenler yerine daha genel bölgeler öğrenilmektedir. Bunun etkisi karar ağaçlarının daha iyi genelleşmesidir. Azınlık sınıfı, asıl büyüklüğünün %100, %200, %300, %400 ve %500'ünde örneklenmiştir. SMOTE algoritması şu şekilde çalışır [26]:

Algorithm SMOTE(T, N, k)
Input: Azınlık sınıflarının sayısı T; yüzdesel SMOTE miktarı N; en yakın komşu sayısı k
Output: (N/100)*T sentetik azınlık örnek sayısı
1. (∗Eğer N, 100'den küçükse, SMOTE edilecek örnek sayısı rastgele seçilir.∗)
2. **if** N<100
3. **then** Rastgele seçilen T azınlık örnek sayısı
4. T=(N/100)∗T
5. N= 100
6. End if
7. N=(int)(N/100)(∗SMOTE edilecek veri 100'ün katı olarak varsayılır.∗)
8. k= En yakın komşu sayısı
9. numattrs= Nitelik sayısı
10. Sample[ ][ ]: Orjinal azınlık sınıf örneklerinin dizisi
11. newindex: ilk değeri sıfır olup sentetik örneklerin sayısını tutar.
12. Synthetic[][]: sentetik örneklerin dizisi (∗Yalnızca her azınlık sınıfı örneği için en yakın komşu hesaplanır.∗)
13. **for** i←1 **to** T

```
14.                 i'ye en yakın komşuları hesaplar ve dizileri nnarray'de saklar.
15.                 Populate(N,i,nnarray)
16.   End for
      Populate(N, i, nnarray)(∗Sentetik örnek üretmek için fonksiyon.∗)
17.   while N≠0
18.                 1 ile k arasında bir rastgele sayı seçilip, nn olarak adlandırılır. Bu adım i'nin en yakın
      komşularından birini seçer.
19.                 for attr←1 to numattrs
20.                       Compute: dif=Sample[nnarray[nn]][attr]−
            Sample[i][attr]
21.                       Compute: gap= 0 ile 1 arasında rastgele sayı.
22.                       Synthetic[newindex][attr]=Sample[i][attr]+gap∗dif
23.                 endfor
24.                 newindex++
25.                 N=N−1
26.   End while
27.   return(∗ Populate fonksiyonunun sonucu ∗)
      Pseudo-Code'un sonu.
```

2.4.2. Yoğun en yakın komşu örnekleme tekniği (Condensed Nearest Neighbour, CNN)

Yoğunlaştırılmış En Yakın Komşu, bir numunenin kaldırılıp kaldırılmayacağına karar vermek için en yakın 1 komşu kuralını kullanır [27]. Algoritma aşağıdaki gibi çalışıyor:
- Bütün azınlık örneklerini bir C setine alın.
- C'ye hedeflenmiş sınıftan(örneklenecek sınıf) ve bu sınıfın diğer tüm örneklerinden bir set S'de bir örnek ekleyin.
- S setine geçin, numuneye göre numune alın ve en yakın bir komşu kuralını kullanarak her numuneyi sınıflandırın.
- Örnek yanlış sınıflandırılmışsa, C'ye ekleyin, başka bir şey yapmayın.
- Eklenecek örnek bulunmayana kadar S üzerinde yeniden düzenleyin.

2.4.3. Rastgele örnekleme tekniği (Random Undersampling Technique, RUS)

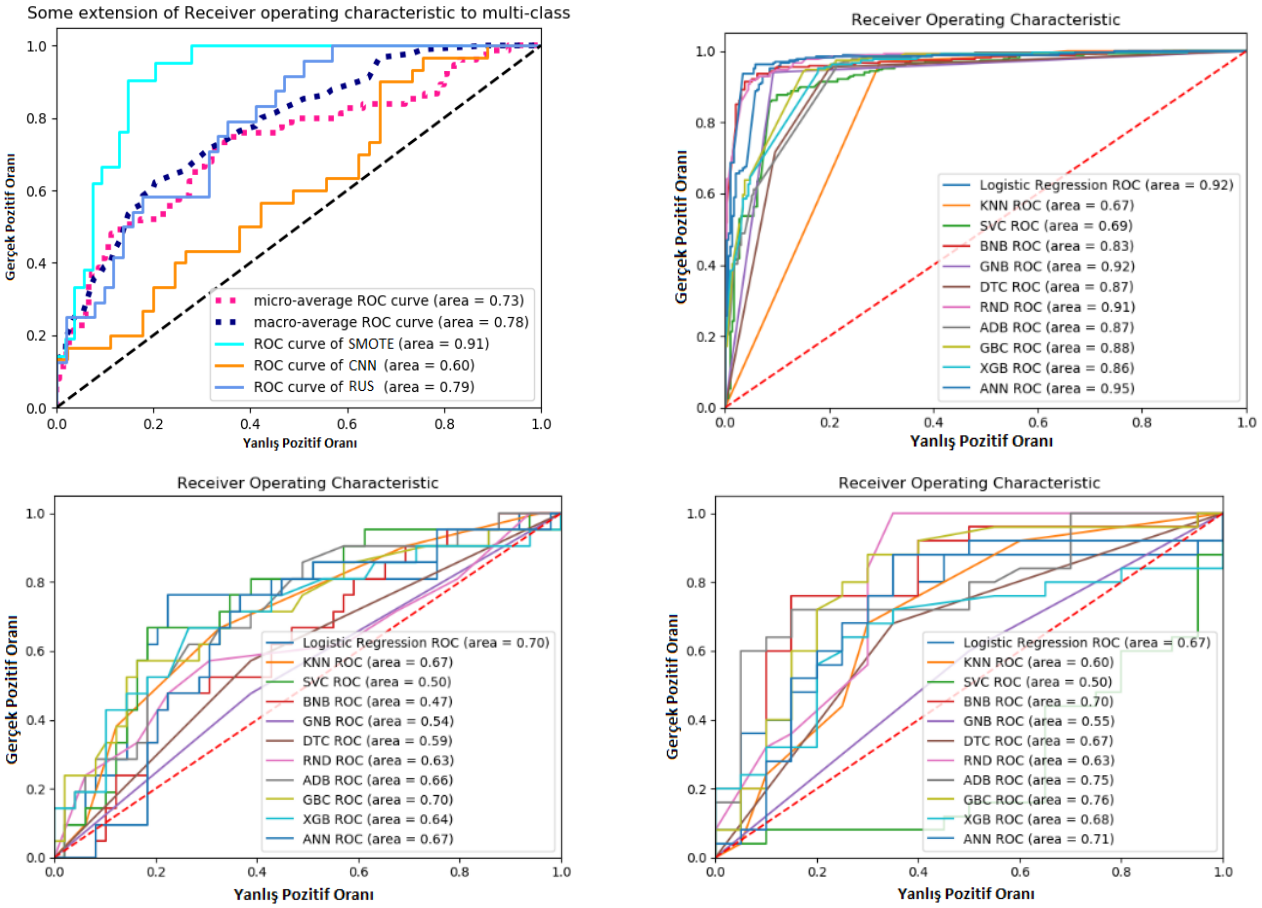
Teknik, numuneleri rastgele bir şekilde sınıftan çıkarır, yerine koyar ya da koymaz. Bu, veri setindeki dengesizliği hafifletmek için kullanılan en eski tekniklerden biridir, ancak sınıflandırıcının varyansını artırabilir ve potansiyel olarak yararlı veya önemli örnekleri atabilir [28].

3. Bulgular

4203 adet SMS'in yer aldığı veri seti üzerinde yapılan çalışmanın sonuçları Tablo 2'de verilmiştir. OACC değerlerine göre yapılan sınıflandırmalar sonucunda SMOTE tekniğinde en iyi sonucu veren sınıflandırma %80.1 ile lojistik regresyon olurken, bu sınıflandırmayı %79.2 ile ANN izlemiştir. RUS tekniği kullanılarak yapılan sınıflandırma sonucunda en iyi sonucu %73.8 ile Lojistik Regresyon verirken, bu sınıflandırmayı %73.6 ile Rassal Orman izlemiştir. CNN tekniğinde ise en iyi sınıflandırma %62.1 ile XGBoost sınıflandırma olmuştur. Bu sınıflandırmayı %61.5 ile Rassal Orman izlemiştir. Yeniden örnekleme tekniklerinin diğer Makine Öğrenmesi teknikleri ile verdikleri doğru sınıflandırma oranları Tablo 2'de detaylı olarak gösterilmiştir. Burada Recall, o teknik ile doğru olarak tahmin edilenlerinin, tüm o teknikteki doğrulara oranını ifade eder. Specificity, o teknik ile yanlış olarak tahmin edilenlerinin, tüm o teknikteki yanlışlara oranını ifade eder. Precision ise doğru olduğu bilinen gözlemlerin doğru tahmin edilenlerinin doğru olduğu bilinenlere oranıdır. SMOTE, RUS ve CNN tekniklerinin her bir Makine Öğrenmesi tekniği ile ayrı ayrı vermiş oldukları Recall, Specificity ve Precision sonuçları detaylı olarak Tablo 3'de verilmiştir.

Tablo 3. Çalışmanın sonucu

(%)		LR	KNN	SVM	BNB	GNB	DT	RF	ADB	GB	XGB	ANN
Specificity (Spam)	RUS	83.59	97.54	98.77	94.15	85.54	87.38	88.62	87.38	92.31	92.62	87.38
	CNN	83.59	90.40	99.69	87.31	72.45	78.02	85.76	84.83	86.69	88.54	79.88
	SMOTE	93.44	79.64	94.55	94.96	69.70	81.71	81.57	91.44	94.96	93.44	82.82
Specificity (Normal)	RUS	83.71	49.85	52.51	76.99	81.12	87.32	86.73	78.17	79.35	75.81	88.20
	CNN	83.71	44.86	100.00	74.94	75.69	86.47	87.72	77.69	85.46	85.21	84.46
	SMOTE	84.53	79.04	65.57	74.77	88.60	93.66	93.80	72.49	76.76	75.41	94.08
Specificity (Reklam)	RUS	70.08	93.64	93.03	87.88	89.70	83.03	85.15	88.79	85.76	86.97	83.64
	CNN	70.08	84.85	7.95	78.41	76.89	65.91	64.02	70.08	65.15	65.53	69.70
	SMOTE	91.98	96.62	91.49	91.21	95.01	91.28	92.69	89.87	89.80	90.58	92.12
Recall (Spam)	RUS	66.46	90.70	95.35	86.03	72.51	75.15	77.71	74.69	82.27	82.61	76.44
	CNN	66.46	68.69	95.45	71.53	51.89	57.99	68.06	65.73	69.29	72.18	62.43
	SMOTE	84.03	63.17	81.46	85.01	59.50	70.02	70.40	75.10	85.10	81.30	71.58
Recall (Normal)	RUS	45.38	46.20	47.73	63.72	68.00	74.40	72.56	63.73	65.17	62.04	75.00
	CNN	45.38	24.66	-	40.83	34.90	46.53	48.96	40.27	46.79	48.70	39.81
	SMOTE	74.11	68.15	57.44	64.28	70.32	84.28	84.79	62.30	66.11	64.83	85.34
Recall (Reklam)	RUS	63.43	77.89	77.67	72.60	73.02	65.85	70.66	71.76	69.68	69.93	66.87
	CNN	63.43	60.78	48.41	68.33	61.64	59.64	62.45	60.70	62.30	62.86	63.13
	SMOTE	83.81	88.38	78.99	81.03	86.19	81.95	84.31	76.59	78.83	79.32	83.88
Precision (Spam)	RUS	61.76	45.35	47.67	68.02	72.09	72.09	75.00	70.35	67.44	66.28	77.33
	CNN	61.76	40.00	12.35	60.59	56.47	57.65	57.65	55.29	57.06	56.47	63.53
	SMOTE	72.67	73.55	50.44	60.17	93.75	89.97	92.30	54.36	60.61	60.03	91.13
Precision (Normal)	RUS	57.45	92.41	93.04	86.71	86.08	79.11	75.32	82.28	82.91	84.81	75.95
	CNN	57.45	76.60	0.00	73.40	55.32	50.00	50.00	63.83	54.26	59.57	43.62
	SMOTE	84.60	85.69	88.83	86.78	51.63	64.99	66.08	86.92	86.65	86.65	65.80
Precision (Reklam)	RUS	59.83	44.31	47.90	63.47	55.09	64.67	70.66	56.29	64.67	59.88	65.27
	CNN	59.83	27.07	99.56	53.71	42.79	58.08	69.00	53.28	66.38	67.25	59.83
	SMOTE	82.52	51.05	63.64	74.69	61.96	78.74	78.18	65.87	75.52	71.89	81.54
OACC	RUS	**73.84**	59.96	62.17	72.43	70.82	71.83	73.64	69.42	71.43	70.02	72.84
	CNN	60.04	40.97	50.51	59.84	49.90	56.39	61.46	55.98	60.85	**62.07**	58.01
	SMOTE	**80.07**	70.19	68.04	74.17	68.65	77.63	78.57	69.40	74.54	73.14	79.22



Şekil 1. Genel ve SMOTE, CNN, RUS tekniklerinin ROC grafikleri

Tablo 4. SMOTE tekniğinin en başarılı algoritmasının confusion matrisi

SMOTE		Gerçek			Doğruluk (%)
		Reklam	Normal	Spam	
Tahmin	Reklam	590	63	62	82.52
	Normal	80	621	33	84.60
	Spam	34	154	500	72.67
					80.07

Tablo 5. CNN tekniğinin en başarılı algoritmasının confusion matrisi

CNN		Gerçek			Doğruluk (%)
		Reklam	Normal	Spam	
Tahmin	Reklam	154	42	33	67.25
	Normal	34	56	4	59.57
	Spam	57	17	96	56.47
					62.07

Tablo 6. RUS tekniğinin en başarılı algoritmasının confusion matrisi

RUS		Gerçek			Doğruluk (%)
		Reklam	Normal	Spam	
Tahmin	Reklam	105	39	23	62.87
	Normal	21	132	5	83.54
	Spam	25	17	130	75.58
					73.84

4. Tartışma ve Sonuç

Araştırma sonucunda, CNN ve RUS teknikleriyle yeniden örnekleme yapılan veri setlerindeki birim sayılarının SMOTE tekniğine göre daha az olması başarı oranına yansıdığı düşünülmektedir. Bu çalışma sonucunda örnekleme tekniklerinden SMOTE tekniğinin örnekleme tekniklerinden daha iyi metin sınıflandırması yaptığı görülmektedir. Ek olarak, bazı algoritmaların örnekleme ve rastgele örnekleme teknikleri uygulanırken ezberleme (overfitting) yaptığı görülmüştür. Bu çalışmada yeniden örnekleme teknikleri detaylı incelenmiş ve

hangi tekniğin ne gibi artısı veya eksisi olduğu görülmüştür. Gelecekte ise sosyal medyaya veya forumlara yapılan kullanıcı yorumlarının sınıflandırmasında verinin normalleştirme aşaması üzerinde detaylı çalışma yapılması planlanmaktadır. Çünkü; günümüzde sosyal medyayı yoğun olarak kullanan genç nüfusun kullandığı kısaltmalar ve dil itibariyle analiz yapılırken ciddi zorluklar çekildiği görülmüş ve o alanda çalışma yapılması düşünülmektedir.

Kaynakça

[1] Tantuğ, A. C. 2016. Metin Sınıflandırma. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5(2).

[2] Chaffar, S., Inkpen, D. 2011. Using a heterogeneous dataset for emotion analysis in text. Butz C., Lingras P. (eds) Advances in Artificial Intelligence. AI 2011. Lecture Notes in Computer Science, vol 6657. Springer, Berlin, Heidelberg; pp. 62-71.

[3] Tüfekci, P., Uzun, E., & Sevinç, B. 2012. Text classification of web based news articles by using Turkish grammatical features. In 2012 20th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[4] Amasyalı, M. F., & Yıldırım, T. 2004. Otomatik haber metinleri sınıflandırma. SIU 2004, 224-226.

[5] Amasyalı, M. F., & Diri, B. 2006. Automatic Turkish text categorization in terms of author, genre and gender. In International Conference on Application of Natural Language to Information Systems (pp. 221-226). Springer, Berlin, Heidelberg.

[6] Yildiz, H. K., Gençtav, M., Usta, N., Diri, B., & Amasyali, M. F. 2007. A new feature extraction method for text classification. In 2007 IEEE 15th Signal Processing and Communications Applications (pp. 1-4). IEEE.

[7] Güven, A., Bozkurt, Ö. Ö., & Kalıpsız, O. 2006. Advanced Information Extraction with n-gram based LSI. In Proceedings of World Academy of Science, Engineering and Technology (Vol. 17, pp. 13-18).

[8] Güran, A., Akyokuş, S., Bayazıt, N. G., & Gürbüz, M. Z. 2009. Turkish text categorization using n-gram words. In Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2009) (pp. 369-373).

[9] Vapnik, V. The nature of statistical learning theory. Springer, 2nd edition, 1995; New York, USA. pp: 32-40.

[10] Müller, K.R., Smola, A., Ratsch, G., Sch¨olkopf, B., Kohlmorgen, J., Vapnik, V. 1997. Predicting time series with support vector machines. International Conference on Artificial Neural Networks 1997; Springer, Berlin, Heidelberg, pp. 999-1004.

[11] Schlögl, A., Lee, F., Bischof, H., Pfurtscheller, G. 2005. Characterization of four-class motor imagery EEG data for the BCI- competition. Journal of neural engineering 2005; 2(4): L14. doi: 10.1088/1741-2560/2/4/L02

[12] Schwarm, S.E., Ostendorf, M. 2015. Reading level assessment using support vector machines and statistical language models. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics 2015; Association for Computational Linguistics, pp. 523-530. doi: 10.3115/1219840.1219905

[13] Friedl, M.A., Brodley, C.E. 1997. Decision tree classification of land cover from remotely sensed data. Remote sensing of environment 1997; 61(3): pp. 399-409. doi: 10.1016/S0034-4257(97)00049-7

[14] Petkovic, D., Altman, R., Wong, M., Vigil, A. 2018. Improving the explainability of Random Forest classifier–user centered approach. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing 2018; Vol. 23. NIH Public Access. pp. 204-215. doi: 10.1142/9789813235533 0019

[15] Piras, P., Sheridan, R., Sherer, E.C., Schafer, W., Welch, C.J., Roussel, C. 2018. Modeling and predicting chiral stationary phase enantioselectivity: An efficient random forest classifier using an optimally balanced training dataset and an aggregation strategy. Journal of separation science; 41(6): pp. 1365-1375. doi: 10.1002/jssc.201701334

[16] Hu, J., Min, J. 2018 Automated detection of driver fatigue based on EEG signals using gradient boosting decision tree model. Cognitive Neurodynamics; pp. 431-440. doi: 10.1007/s11571-018-9485-1

[17] Yang, L., Zhang, X., Liang, S., Yao, Y., Jia, K., Jia, A. 2018. Estimating Surface Downward Shortwave Radiation over China Based on the Gradient Boosting Decision Tree Method. Remote Sensing; 10(2): 185. doi: 10.3390/rs10020185

[18] Monisha, A., Christina, S.S., Santiago, N. 2018. Decision Support System for a Chronic Disease-Diabetes. International Journal of Computer & Mathematical Sciences(IJCMS); ISSN 2347-8527, Volume 7, Issue 3, pp: 126-131.

[19] Celik, O., Osmanoglu, U.O. 2019. Comparing to Techniques Used in Customer Churn Analysis. Journal of Multidisciplinary Developments, 4(1), 30-38.

[20] Estabrooks, A. 2000. A combination scheme for inductive learning from imbalanced data sets, Diss. DalTech.

[21] Estabrooks, A., Jo, T., Japkowicz, N. 2004 A multiple resampling method for learning from imbalanced data sets. Compu tational intelligence; 20(1): pp. 18-36. doi: 10.1111/j.0824-7935.2004.t01-1-00228.x
[22] Sun, Y., Kamel, M. S., Wong, A. K., & Wang, Y. 2007. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition, 40(12), 3358-3378.
[23] https://www.researchgate.net/publication/310799885_Generalized_Confusion_Matrix_for_Multiple_Classes (Erişim Tarihi: 21/10/2020)
[24] https://github.com/scikit-learn-contrib/imbalanced-learn (Erişim Tarihi: 21/10/2020)
[25] https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/smote (Erişim Tarihi: 21/10/2020)
[26] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
[27] https://imbalanced-learn.readthedocs.io/en/stable/under_sampling.html#condensed-nearest-neighbors (Erişim Tarihi: 21/10/2020)
[28] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905.