

Yarı denetimli makine öğrenmesi yöntemini kullanarak müzik türlerinin tespiti

Alp Kaan TURAN^{1*} , Hüseyin POLAT¹ 

¹Gazi Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 06500, Yenimahalle/ANKARA

Makale Bilgisi

Araştırma makalesi
Başvuru: 30/08/2023
Düzeltilme: 11/10/2023
Kabul: 13/12/2023

Anahtar Kelimeler

Makine öğrenmesi
Yarı denetimli öğrenme
Kendi kendine öğrenme

Article Info

Research article
Received: 30/08/2023
Revision: 11/10/2023
Accepted: 13/12/2023

Keywords

Machine learning
Semi-supervised learning
Self-learning
Self-training

Grafik Özet (Graphical/Tabular Abstract)

Sınıflandırma işlemine ait akış şeması. / Flow chart of the classification process



Figure A: Flow chart diagram / Şekil A: İş akış şeması

Önemli noktalar (Highlights)

- Yarı denetimli öğrenme ile sınıflandırma / Classification with semi-supervised learning
- Kendi kendine öğrenme yönteminin uygulanması / Implementation of the self-training method

Amaç (Aim): Kendi kendine öğrenme yönteminin sınıflandırma performansına etkisinin belirlenmesi. / Determining the effect of self-training method on classification performance.

Özgünlük (Originality): GTZAN müzik türleri veri kümesi üzerinde sekiz farklı sınıflandırıcı model için kendi kendine öğrenme yönteminin katkısı değerlendirilmiştir. / The contribution of the self-training method is evaluated for eight different classifier models on the GTZAN music genres dataset.

Bulgular (Results): Daraltılmış eğitim kümesi ile yapılan ölçümlerle karşılaştırıldığında kendi kendine öğrenme yöntemi, k-en yakın komşu sınıflandırıcı için 0,02-0,04 ve çok katmanlı yapay sinir ağı sınıflandırıcı için 0,05-0,09 doğruluk artışı sağlanmıştır. Bu iki sınıflandırıcı üzerinden, destekleyici yöntemlerin katkısı değerlendirilmiştir. Eklenen bir dengeleme filtresiyle kendi kendine öğrenme yönteminin doğruluk oranında yaklaşık 0,01 oranında ek bir artış sağlanmıştır. Diğer sınıflandırıcılarla yapılan deneylerde, kendi kendine öğrenmenin sınıflandırma performansını değiştirmedeği ya da düşürdüğü görülmüştür. / Compared to the measurements with the reduced training set, the self-learning method improved accuracy by 0.02-0.04 for the k-nearest neighbor classifier and 0.05-0.09 for the multilayer neural network classifier. On these two classifiers, the contribution of the supplementary methods was evaluated. With the addition of a stabilization filter, an additional increase of about 0.01 in the accuracy of the self-learning method was achieved. Experiments with other classifiers showed that self-learning did not change the classification performance or decreased.

Sonuç (Conclusion): Kullanılan veri kümesi ve uygulanan sınıflandırıcı türüne bağlı olarak, modelin kendi kendine eğitim yöntemi ile desteklenmesi sınıflandırma başarısına katkı sağlayabilmektedir. / Depending on the dataset used and the type of classifier applied, the self-training of the model can contribute to the classification success.



Yarı Denetimli Makine Öğrenmesi Yöntemini Kullanarak Müzik Türlerinin Tespiti

Alp Kaan TURAN^{1*} , Hüseyin POLAT¹

¹Gazi Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 06500, Yenimahalle/ANKARA

Makale Bilgisi

Araştırma makalesi
Başvuru: 30/08/2023
Düzeltilme: 11/10/2023
Kabul: 13/12/2023

Anahtar Kelimeler

Makine öğrenmesi
Yarı denetimli öğrenme
Kendi kendine öğrenme

Öz

Denetimli makine öğrenmesi yöntemleri ile başarılı sonuçlar elde etmek için yüksek miktarda etiketli veriye ihtiyaç duyulur. Etiketleme, zaman ve işgücü açısından maliyetli bir işlemdir. Etiketleme için gereken zaman ve işgücünü azaltmak için çeşitli yöntemlerden faydalanılır. Etiketli verinin etiketsiz veriyle birlikte kullanıldığı yarı denetimli öğrenme de bu yöntemlerden biridir. Bu çalışmada, GTZAN veri kümesindeki müzik türlerinin tespitinde bir yarı-denetimli öğrenme yöntemi olan kendi kendine öğrenmenin katkısı değerlendirilmiştir. Müzik türlerini tespit etmek için kendi kendine öğrenme yöntemi sırasıyla Naive Bayes, K-En Yakın Komşu, Karar Ağacı, Destek Vektör Makinesi, Lojistik Regresyon, Çok Katmanlı Yapay Sinir Ağı, XGBoost ve Rastgele Orman sınıflandırıcılara uygulanmıştır. Çalışmada kullanılan veri kümesi, dört farklı oranda (%4, %8, %16, %32) etiketli veri, %10 test verisi ve kalan kısmı etiketsiz veri olacak şekilde ayrılmıştır. Her bir oran için daraltılmış etiketli, etiketsiz kümeye sözde etiket atayarak ve test dışındaki bölüm tamamen etiketli olacak şekilde üç farklı eğitim kümesi oluşturulmuştur. Bu kümeler ile makine öğrenmesi modelleri eğitilmiş ve daha önceden ayrılmış test verisi ile model performansları sınanmıştır. Çalışmada, kendi kendine öğrenme yönteminin etkisi doğruluk ve F1-Skor ölçütleri yönünden değerlendirilmiştir. Daraltılmış eğitim kümesi ile yapılan ölçümlerle karşılaştırıldığında, kendi kendine öğrenme yöntemi, k-en yakın komşu sınıflandırıcı için 0,02-0,04 ve çok katmanlı yapay sinir ağı sınıflandırıcı için 0,05-0,09 doğruluk artışı sağlanmıştır. Bu iki sınıflandırıcı üzerinden, destekleyici yöntemlerin katkısı değerlendirilmiştir. Eklenen bir dengeleme filtresiyle kendi kendine öğrenme yönteminin doğruluk oranında yaklaşık 0,01 oranında ek bir artış sağlanmıştır. Diğer sınıflandırıcılarla yapılan deneylerde, kendi kendine öğrenmenin sınıflandırma performansını değiştirmedeği ya da düşürdüğü görülmüştür.

Music Genre Detection Using Semi-Supervised Machine Learning

Article Info

Research article
Received: 30/08/2023
Revision: 11/10/2023
Accepted: 13/12/2023

Keywords

Machine learning
Semi-supervised learning
Self-learning
Self-training

Abstract

Supervised machine learning methods require large amounts of labeled data to achieve successful results. Labeling is a costly process in terms of time and labor. Various methods are used to reduce the time and labor required for labeling. Semi-supervised learning, where labeled data is used together with unlabeled data, is one of these methods. In this study, we evaluate the contribution of self-learning, a semi-supervised learning method, in the detection of music genres in the GTZAN dataset. The self-learning method was applied to Naive Bayes, K-Nearest Neighbor, Decision Tree, Support Vector Machine, Logistic Regression, Multilayer Neural Network, XGBoost and Random Forest classifiers respectively. The dataset used in the study was divided into four different proportions (4%, 8%, 16%, 32%) of labeled data, 10% test data and the remainder unlabeled data. For each proportion, three different training sets were created: narrowed labeled, by assigning pseudo-labels to the unlabeled set, and with the non-test portion fully labeled. Machine learning models were trained with these sets and the model performances were tested with the pre-separated test data. In the study, the effect of the self-learning method was evaluated in terms of accuracy and F1-Score. Compared to the measurements with the reduced training set, the self-learning method improved accuracy by 0.02-0.04 for the k-nearest neighbor classifier and 0.05-0.09 for the multilayer neural network classifier. On these two classifiers, the contribution of the supplementary methods was evaluated. With the addition of a stabilization filter, an additional increase of about 0.01 in the accuracy of the self-learning method was achieved. Experiments with other classifiers showed that self-learning did not change the classification performance or decreased.

1. GİRİŞ (INTRODUCTION)

Yapay zekanın bir dalı olan makine öğrenmesi, el alınan veriyi en iyi şekilde genelleyen modelleri ortaya çıkarmak için kullanılan yöntemleri içerir. Makine öğrenmesi yöntemleriyle, bilinen veri örnekleri eğitilen modelle, örnekler arasındaki ilişkinin ortaya çıkarılması ya da yeni girdilere ait sonuçların tahmin edilmesi amaçlanır. Makine öğrenmesi, denetimli öğrenme (*supervised learning*) ve denetimsiz öğrenme (*unsupervised learning*) olarak adlandırılan iki temel kategoriye sahiptir. Denetimli öğrenme için kullanılan veri kümesinde, her bir örnek için n adet girdi özneliği ($x_1, x_2, x_3 \dots x_n$) ve genellikle bir hedef özneliği (y) bulunur. Bu tür veri kümeleri, içerdiği örnekler etiket olarak da adlandırılan hedef özneliğine sahip olması nedeniyle “etiketli veri kümesi” olarak adlandırılır. Eğitim veri kümesi ile eğitilen model kullanılarak yeni girdi örnekleri için hedef özneliğinin değeri tahmin edilmeye çalışılır. Bu kategorideki yöntemler genellikle regresyon ve sınıflandırma işlemlerinde kullanılır.

Denetimsiz öğrenme yönteminde kullanılan veri kümesindeyse sadece n adet girdi özneliği ($x_1, x_2, x_3 \dots x_n$) bulunur ve hedef özneliği bulunmaz. Bu sebeple, denetimsiz öğrenmede kullanılan veri kümesi “etiketsiz veri kümesi” olarak adlandırılır. Denetimsiz öğrenmede, veri örnekleri benzerliklerine göre kümeleme işlemine tabi tutulur.

Denetimli ve denetimsiz öğrenme kategorileri dışında, yarı denetimli öğrenme (YDÖ, *semi-supervised learning*) olarak adlandırılan başka bir kategori de bulunur. YDÖ, kavramsal olarak denetimli ve denetimsiz öğrenme arasında konumlanır. YDÖ üzerine yapılan çalışmalarda genellikle sınıflandırmaya odaklanılmıştır. Performansı arttırmak için, sınırlı miktarda etiketli veri ile birlikte etiketlenmemiş veriden de yararlanılmaya çalışılır [1].

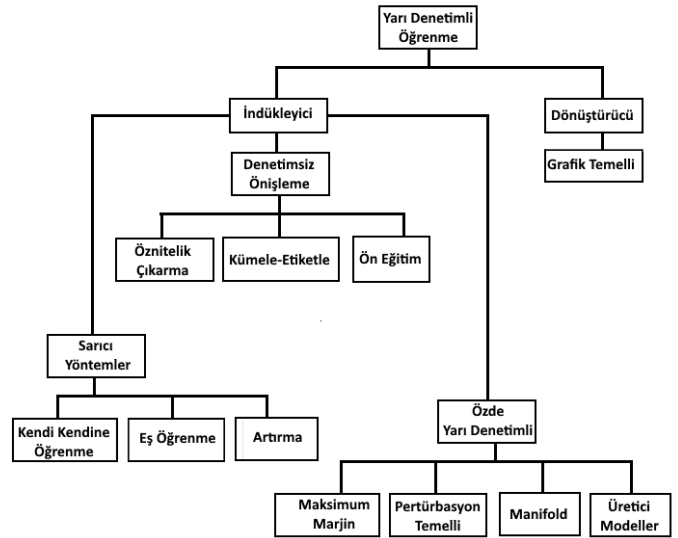
YDÖ, etiketlenmiş verinin sınırlı olduğu durumlarda performanslı bir sınıflandırma yapabilmek için kullanılan etkili bir stratejidir [2]. YDÖ’de, model etiketli ve etiketsiz veri birlikte kullanılarak eğitilir ve yeni veri örnekleri için hedef özneliğinin değeri tahmin edilir [3].

Sınıflandırma bağlamında YDÖ’deki temel varsayım, aynı yüksek yoğunluk bölgesinde bulunan iki veri örneğinin büyük olasılıkla aynı sınıf etiketine sahip olması gerektiğidir. Bu kabul, düzgünlük varsayımı (*smoothness assumption*) olarak adlandırılır [4]. Düzgünlük varsayımına göre veri kümesindeki iki veri örneğinin girdi

öznelikleri (x, x') birbirine yakınsa, bu örneklere ait hedef özneliklerinin sınıf etiketleri de (y, y') aynı olmalıdır. Düzgünlük varsayımı, aynı zamanda geçişli olarak da uygulanabilir. X_1 , sınıf etiketine sahip bir veri örneği olsun. Ayrıca sınıf etiketine sahip olmayan X_2 ve X_3 veri örnekleri mevcut olsun. Veri örnekleri uzayında X_2 veri örneği, X_1 veri örneğine ve X_3 veri örneği, X_2 veri örneğine yakınsa, geçişli yayılımdan dolayı, X_1 ’e yakın olmasa bile X_3, X_1 veri örneği ile aynı sınıf etiketine sahip olacaktır [2].

YDÖ’de indükleyici (*inductive*) ve dönüştürücü (*transductive*) olmak üzere iki ana yöntem grubu mevcuttur [5]. İndükleyici yöntemlerle, etiketli veri yardımıyla etiketsiz veriye sözde etiket (*pseudo label*) atanır. Dönüştürücü yöntemlerle, etiketli ve etiketsiz veri birlikte kullanılarak graf tabanlı bir model oluşturulur [3]. YDÖ yöntemleri Şekil 1’deki gibi sınıflandırılır.

İndükleyici yöntemler kendi içinde, denetimsiz ön işleme (*unsupervised preprocessing*), özde yarı denetimli (*intrinsically semi-supervised*) ve sarıcı (*wrapper*) yöntemler olarak üçe ayrılabilir. Denetimsiz ön işleme (*unsupervised preprocessing*) yöntem grubu öznelik çıkarma (*feature extraction*), küme-etiketle (*cluster-then-label*), ön eğitim (*pre-training*) gibi yöntemleri kapsar. Bu yöntemlerde, sözde etiket atanan örnekler eğitim aşamasında kullanılmaz; etiketli veri kullanılarak etiketsiz veri kümeleme işlemine tabi tutulur. Bu işlemden sonra oluşan kümelere ve dolayısıyla içerdikleri veri örneklerine sınıf etiketi atanarak model oluşturulur [6].



Şekil 1. YDÖ yöntemlerinin sınıflandırılması
(Classification of semi-supervised learning methods) [1]

Maksimum-marj (*maximum-margin*), manifold (*manifolds*), pertürbasyon temelli (*perturbation-based*), üretici modeller (*generative models*) gibi teknikleri içeren özde yarı denetimli (*intrinsically semi-supervised*) yöntemler, sınıflandırıcının amaç ya da iyileştirme fonksiyonunun etiketsiz veriyi de içerecek şekilde genişletilmesine dayanır [7]. Özde yarı denetimli yöntemlere örnek olarak, etiketsiz veriyi de kapsayacak şekilde aralığın maksimize edildiği yarı denetimli destek vektör makinesi (*semi-supervised support vector machines-S3VM*) verilebilir [8].

Sarıcı yöntemler, en eski ve iyi bilinen YDÖ yöntemlerindedir [9]. Sarıcı yöntemler kendi arasında kendi kendine öğrenme (KKÖ, *self-learning*), eş öğrenme (*co-training*) [10] ve artırma (*boosting*) olarak üçe ayrılır. Bu yöntemlerde ana ilke, etiketli veri kullanarak eğitilen sınıflandırıcı yardımıyla etiketsiz veri örneklerine sözde etiket (sınıf etiketi) atanmasıdır. Etiket atanacak örnek kalmayınca kadar, eğitim ve sözde etiket atama süreci döngüler halinde tekrarlanır. Eş öğrenmenin KKÖ'den farkı, birden fazla modelin kullanılması ve bu modellere ait sözde etiket atanmış örneklerin diğer modelin eğitim kümesine eklenmesidir [11], [12]. KKÖ yöntemi, en temel sözde etiket verme yaklaşımıdır [6].

Bu çalışmada, KKÖ'nün çeşitli sınıflandırıcılara olan etkisi GTZAN ses veri kümesi [13] üzerinde değerlendirilmiştir. Literatürdeki diğer çalışmalardan farklı olarak, KKÖ'nün etkisi, herhangi bir ek yöntemle desteklenmeden sekiz farklı sınıflandırıcı model üzerinde test edilmiştir. Daha sonra, KKÖ ek bir seçim yöntemi ile güçlendirilerek en başarılı iki sınıflandırıcı üzerinde testler yürütülmüş, sonuçlar paylaşılmıştır. Bu sayede elde edilen deneysel sonuçlarla KKÖ'nün yalın ve destekleyici bir yöntemle kullanımının etkisi, olumlu ve olumsuz yönleriyle birlikte ortaya koyulmuştur.

KKÖ yöntemi, literatürde *kendi kendine eğitim* (*self-training*), *kendi kendine etiketleme* (*self-labeling*), *karar yönelimli öğrenme* (*decision-directed learning*) gibi kavramlarla yer bulmuştur [2]. KKÖ yöntemi, bir sınıflandırıcı tarafından belirli bir eşik değerinin üstündeki olasılıklar için etiketsiz veri örneklerine sözde etiket atanması prensibine dayanır. Böylece, etiketlenmiş veri örnekleri ile birlikte sözde etiket verilmiş veri örneklerinin bir arada kullanılması, aynı sınıfa ait bölgelerin daha etkili bir şekilde belirlenmesini sağlar [4]. KKÖ yönteminde, genellikle aşağıdaki adımlar izlenir:

- 1) Etiketli veriler kullanılarak bir sınıflandırıcı eğitilir.
- 2) Eğitilmiş modelle etiketlenmemiş veri örneklerine sözde etiket atanır.
- 3) Etiketli ve sözde etiket atanmış veri örnekleri birlikte kullanılarak sınıflandırıcı yeniden eğitilir.
- 4) Etiket atanabilecek etiketsiz veri kalmayınca ya da etiketsiz verinin tümü etiketleninceye kadar 2. ve 3. adımları tekrarlanır.

KKÖ yönteminin işleyişi, Algoritma 1'de gösterilmiştir. Etiketli verilerin bulunduğu veri kümesi X_L , etiketsiz verilerin olduğu veri kümesi X_U ve başlangıç durumunda boş olan sözde etiket atanmış veri kümesi X_{se} ile temsil edilir. İlk durumda, X_L kümesindeki etiketli veri ile model eğitilir. Eğitilen model kullanılarak, X_U kümesinde yer alan, belirli bir eşik değerinin üstünde sınıflandırma doğruluğuna sahip veri örneklerine sözde etiket atanır. Bu veri örnekleri X_{se} kümesine taşınır ve X_U kümesinden çıkarılır. Sonraki aşamada döngüsel olarak X_L ve X_{se} kümelerinin birleşimi ($X_L \cup X_{se}$) kullanılarak mevcut model tekrar eğitilir. X_U kümesindeki kalan veri örneklerine aynı kabuller doğrultusunda sözde etiket atanır. Sözde etiket atanmış veri örnekleri X_{se} kümesine taşınır, aynı zamanda ilgili veri örnekleri X_U kümesinden de çıkarılmış olur. Bu işlem X_U kümesi boşalana ve X_{se} kümesine eklenen yeni veri örneği kalmayınca kadar devam eder.

Algoritma 1. KKÖ algoritması (Self-learning algorithm) [14]

Girdi: $X_L =$ etiketli veri kümesi
 $X_U =$ etiketsiz veri kümesi
 $X_{se} = \{\}$ sözde etiket atanmış veri kümesi, başta boş

döngü = 0

tekrarla:
 $X_L \cup X_{se}$ ile sınıflandırıcıyı eğit
 X_U 'da marj içinde kalan veriye sözde etiket ver
Sözde etiket verilen örnekleri:
 X_U kümesinden çıkar,
 X_{se} kümesine ekle

döngü ++, döngü sayısını artır.
 $X_U = \{\}$ etiketsiz veri kümesi boşsa ya da döngüde sözde etiket verilen örnek yoksa çık
değilse tekrarla

Buradaki kritik noktalardan biri, her döngü sırasında sözde etiket verilecek alt kümenin nasıl tespit edileceği konusudur. KKÖ modeli, her döngüde belirli bir etiketsiz veri alt kümesini seçmelidir. Aksi takdirde, tüm etiketsiz veri örneklerine ilk döngüde etiket atanır. Bu da, modelin, sözde etiket atama işlemi için, etiketli veri ile yapılan sınıflandırmayla aynı performansa sahip

olmasına yol açar [15]. Bu noktada, her bir sınıf için güvenilir eşik değerlerinin belirlenmesi gerekmektedir. İkili sınıflandırmada, sözde etiket atama işleminde hedef değer seçimi Eşitlik 1'e göre gerçekleştirilir.

$$\tilde{y} = \begin{cases} +1, & \text{if } f(x, +1) \geq \theta, \\ -1, & \text{if } f(x, -1) \geq \theta, \end{cases} \quad (1)$$

\tilde{y} : atanacak sınıf etiketi (sözde etiket), θ : eşik değeri,

x : girdi öznelikleri, $f(x, n)$: sınıflandırıcının ürettiği olasılık değeri

Eşik değer (θ), yanlış etiketleme ihtimalini azaltacak şekilde yüksek belirlenmelidir. Öte yandan, çok yüksek bir eşik değeri belirlenmesi halinde her döngüde sözde etiketleme işlemi için ayrılan etiketsiz alt veri kümesi küçülecektir. Bu durumda, sözde etiket atanan veri örneği sayısı azalacaktır. İkili sınıflandırmada, en uygun eşik değerini belirlemek için kullanılan yöntemlerden biri, tahmin edilen pozitif ve negatif sınıfların olasılık değerlerinin ortalamasının alınmasıdır [16].

İkili sınıflandırma dışında, çoklu sınıflandırma için de, uygun eşik değerinin belirlenmesi konusunda çeşitli çalışmalar mevcuttur. Bunlardan biri Lee ve diğerlerinin çalışmasıdır [17]. Çalışmada, çoklu sınıflandırma için uygun eşik değeri yapay sinir ağı modeli kullanılarak belirlenir. Tür ve diğerlerinin çalışmasının [16] çok sınıflı veri kümelerine uygulandığı başka bir çalışmada [18] eşik değer seçilmeden sözde etiket atama işlemi yürütülür. En güvenilir etiketsiz veri oranına (p) başta 0,20 değeri atanır ve maksimum 0,50 oluncaya kadar her döngüde bu oran 0,05 artırılır. Cascante-Bonilla ve diğerleri tarafından yapılan çalışmada [19], sözde etiket atama için *müfredat öğrenmesi (curriculum learning)* modeli önerilmiştir. Müfredat öğrenmesi, çıkarım yapılması kolay veriden zora doğru modelin eğitilmesi esasına dayanır. İlgili çalışmada etiketsiz verilerin tümü etiketlenene kadar döngüler halinde işlem tekrarlanır. Karşılaştırma için 1 ile 0 arasında 0,2'lik adımlarla azalan yüzdelik skor eşik değeri kullanılır. Her döngüde azalan eşik değerinin üzerinde skora sahip etiketsiz veri örneklerine sözde etiket atanır, veri örneği eğitim kümesine eklenir ve model yeniden eğitilir [19]. Bunların dışında, Rastgele Orman ve Adaboost gibi çoğunluk oyu sınıflandırıcılarda, sözde etiketleme işleminde eşik değeri seçimi otomatik olarak yapılır [2].

Eş öğrenme başka bir KKÖ yaklaşımıdır. KKÖ'den farkı, birlikte çalışan en az iki model içermesidir. Eş öğrenme yaklaşımlarından biri *Noisy student* [11] yöntemidir. Bu yöntemde, bir öğretmen bir de öğrenci modeli bulunur. Öğretmen model doğru

etiketlenmiş veri ile eğitilir ve bu model öğrenci modeli eğitmekte kullanılır. Öğrenci sınıflandırıcı etiketsiz veri üzerinde tahminleme yapar ve sonucu öğretmen sınıflandırıcıya geri besler. Başka bir eş öğrenme yöntemi olan *FixMatch* [20] algoritmasında, iki modelden biri doğru etiketli, diğeri gürültü eklenmiş veri ile eğitilir. Doğru etiketli veri ile eğitilen model, gürültü eklenmiş veri ile eğitilen modeli düzeltmek için kullanılır. Entropi kaybı üzerinden veri kümesinin olasılık dağılımı hesaplanarak karşılaştırma yapılır.

Konuyla ilgili çalışmalar kapsamında, *Yılmaz* ve diğerlerinin çalışmasında [21] GTZAN veri kümesi üzerinde *Naive Bayes*, *K-En Yakın Komşu*, *Rastgele Orman*, *Destek Vektör Makinesi*, *Lojistik Regresyon* ve *XGBoost* sınıflandırıcıları kullanılmış; *K-En Yakın Komşu* ve *XGBoost* ile sırasıyla %91,59 ve %91,79 doğruluk oranı elde edilmiştir. *Sigtia* ve *Dixon* tarafından yapılan başka bir çalışmada [22], GTZAN veri kümesi üzerinde derin öğrenme kullanarak yapılan sınıflandırmada %83 doğruluk elde edilmiştir.

Kiranoğlu ve diğerlerinin çalışmasında [23] suç oluşum tahmininde KKÖ'nün katkısını değerlendirilmiştir. Bu kapsamda 2019 yılında İngiltere'de işlenen suçlar veri kümesi üzerinde *Naive Bayes*, *K-En Yakın Komşu*, *Karar Ağacı*, *Destek Vektör Makinesi* sınıflandırıcılar kullanılarak deneysel çalışma yürütülmüştür. *Triguero* ve diğerleri tarafından yapılan başka bir çalışmada [24] KKÖ yönteminin etkisi ölçülmüştür. 60 veri kümesi üzerinden gürültü filtreleme teknikleri ile desteklenmiş KKÖ yönteminin etkililiği değerlendirilmiştir.

Wang ve diğerlerinin çalışmasında [25] çeşitli görüntü ve ses veri kümeleri üzerinde π -*Model*, *Pseudo-Labeling*, *Mean Teacher*, *VAT*, *UDA*, *FixMatch* [20], *Dash*, *CoMatch*, *CRMatch*, *FlexMatch* [26], *AdaMatch*, *SimMatch* YDÖ yöntemlerinin sınıflandırma başarıları ve hata oranları karşılaştırılmıştır.

Zoph ve diğerleri tarafından yapılan çalışmada [27], görüntü veri kümeleri üzerinde veri artırımı, kısıtlı veri ile ön eğitim ve KKÖ yöntemlerinin sınıflandırma başarısına etkileri değerlendirilmiştir.

Li ve *Zhou'nun* çalışmasında [28], KKÖ yöntemi, SETRED adı verilen grafik tabanlı dağılım filtresi algoritmasıyla desteklenmiştir. Çalışmada, yaygın kullanılan on adet veri kümesi üzerinde deneysel çalışmalar yürütülmüştür. Çalışma sonucunda, KKÖ yöntemi tek başına veri kümelerinin sadece altısında model performansı artırdığı, dördünde düşürdüğü görülmüştür. Diğer taraftan, SETRED

ile desteklenen KKÖ yöntemi kullanıldığında, veri kümelerinin sekizinde sınıflandırma performansında artış olurken ikisinde düşmüştür.

Zou ve diğerleri [29] KKÖ'de gürültü kaynaklı yanlış etiket atama problemiyle başa çıkmak, atama işleminin güvenilirliğini artırmak için *CRST* (*confidence regularized self-training*) yöntemini önermişlerdir. Etiket ve model düzenleme içeren yöntem, görüntü sınıflandırmada doğruluk oranı artışı sağlamıştır. Sohn ve diğerleri tarafından yapılan çalışmada [20], KKÖ, modelin genelleme yeteneğini artırmak için entropi tabanlı dengeleme teknikleriyle desteklenmiştir. Çalışmada, *FixMatch* adı verilen algoritma CIFAR-10 [30] görüntü veri kümesine uygulanmıştır. Sadece 250 etiketli veri örneği kullanılarak %94,93 doğruluk oranı elde edilmiştir. Bir başka çalışmada [26], *Fixmatch* algoritmasının türevi olan *FlexMatch* algoritması önerilmiştir. *FlexMatch* algoritmasında, esnek etiketlemeyi sağlayan ve Müfredat Sözcük Etiketleme (*Curriculum Pseudo Labeling*) adı verilen yöntem kullanılmıştır. Bu algoritmanın eğitim süresi *Fixmatch* algoritmasının beşte biri kadardır. Ayrıca, çok az miktarda eğitim verisiyle *Fixmatch*'e göre daha başarılı sonuçlar elde edilmesini sağlamıştır. Yakın zamanda yapılan bir diğer çalışmada [31], *Fixmatch* algoritmasının bir başka türevi *DeFixmatch* algoritması önerilmiştir. Yapılan deneylerde daha yüksek doğruluk oranı elde edilmiştir. Ancak, bu algoritmanın hesaplama yükü ve zaman maliyeti *Fixmatch*'e göre daha yüksektir.

Literatür taraması sonucunda, etiketli verinin yetersiz olduğu durumlarda, KKÖ'nün performans artırıcı yardımcı bir yöntem olarak başarılı uygulamalarının olduğu görülmüştür.

2. MATERYAL VE METOD (MATERIAL AND METHODS)

Tzanetakis ve diğerleri tarafından hazırlanan GTZAN müzik türü veri kümesi [13], 10 farklı türde (*blues, classic, country, disco, hip hop, jazz, metal, pop, reggae, rock*), her tür için 1000 adet olmak üzere toplam 10000 dosyadan oluşur. Her bir dosya, 22050 Hz örnekleme frekansında 16 bit .au formatında kaydedilmiş üçer saniyelik ses verisi içerir. Bu çalışmada, GTZAN veri kümesi tekrar ön işleme tabi tutulmamıştır. *Olteanu* tarafından yapılan çalışmada [32] elde edilen ve 60 öznelikten oluşan veri kümesi kullanılmıştır. Veri kümesindeki öznelikler Tablo 1'de gösterilmektedir.

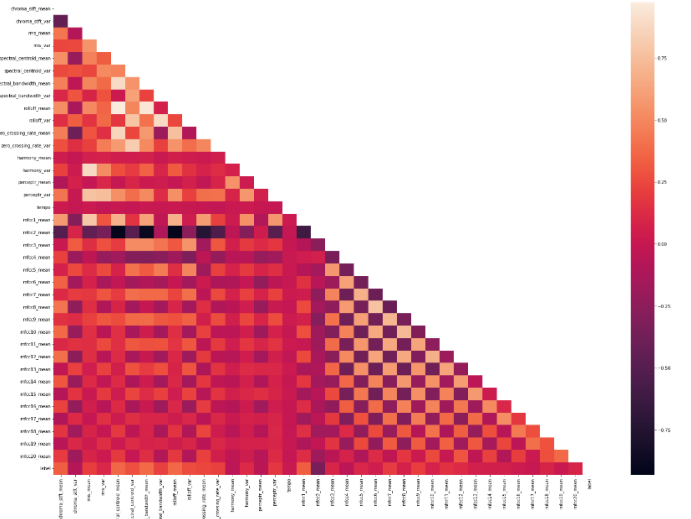
Tablo 1. GTZAN veri kümesine ait öznelikler (Features in GTZAN dataset)

ÖZNEİELİK ADI	MİKTAR
Dosya adı, Saniye cinsinden süre, Tempo, Müzik türü	4
Kroma kısa fourier frekans dönüşümü ortalama ve varyansı, karekök ortalama ve varyansı, spektral ağırlık merkezi ortalaması ve varyansı, bant genişliği ortalaması ve varyansı, kesim frekansı ortalama ve varyansı, sıfır geçiş oranı ortalaması ve varyansı, harmoni ortalaması ve varyansı, algısal oran ortalaması ve varyansı	16
Mel frekans kepstral katsayısı ortalaması ve varyansı (20 Kanal)	40

2.1. Ön İşleme (Preprocessing)

İlk olarak, veri kümesinden kullanılmayacak olan "dosya adı" ve "saniye cinsinden süre" öznelikleri çıkarıldı. Sayısal öznelikler normalizasyon işlemine tabi tutulmuştur. Normalizasyon işlemine min-maks normalizasyon yöntemi kullanılmıştır. Bir özneliğin alabileceği değerlerin en küçüğü $\min(x)$ en büyüğü $\max(x)$ olduğu kabul edilirse, min-maks normalizasyon sonrası oluşan ölçeklendirilmiş x' değeri Eşitlik 2'deki gibi hesaplanır:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2)$$



Şekil 2. GTZAN veri kümesine ait ısı haritası (Heatmap of GTZAN dataset)

Sonrasında, öznelikler için ısı haritası oluşturulmuştur (Şekil-2). 0,80 üzerindeki değerler pozitif, -0,80 altındaki değerler negatif yüksek korelasyonu gösterir. Buna göre, Spektral Ağırlık Merkezi Ortalaması özneliği ile Kesim Frekansı Ortalaması (0,97) ve Sıfır Geçiş Oranı Ortalaması Katsayısı (0,87) öznelikleri arasında görece yüksek bir korelasyon bulunmaktadır. Ayrıca, 2. Kanal Mel Frekans Kepstral Katsayısı özneliği ile Spektral Ağırlık Merkezi Ortalaması (-0,93), bant genişliği Ortalaması (-0,89) ve Kesim Frekansı Ortalaması (-0,92) öznelikleri arasında görece daha yüksek bir negatif korelasyon vardır.

Girdi özneliklerine bağlı olarak hedef sınıf etiketinin dağılımını ortaya çıkarmak için, öznelikler PCA (Temel Bileşen Analizi) yöntemi ile üçe indirildi. Sonrasında, üç farklı açıdan görünecek şekilde dağılım grafiği oluşturuldu. Grafiklerden de anlaşılacağı üzere, bazı sınıflara (*rock*, *country* vb.) ait verilerin homojen bir dağılıma sahip olmadıkları görülmüştür (Şekil 3).

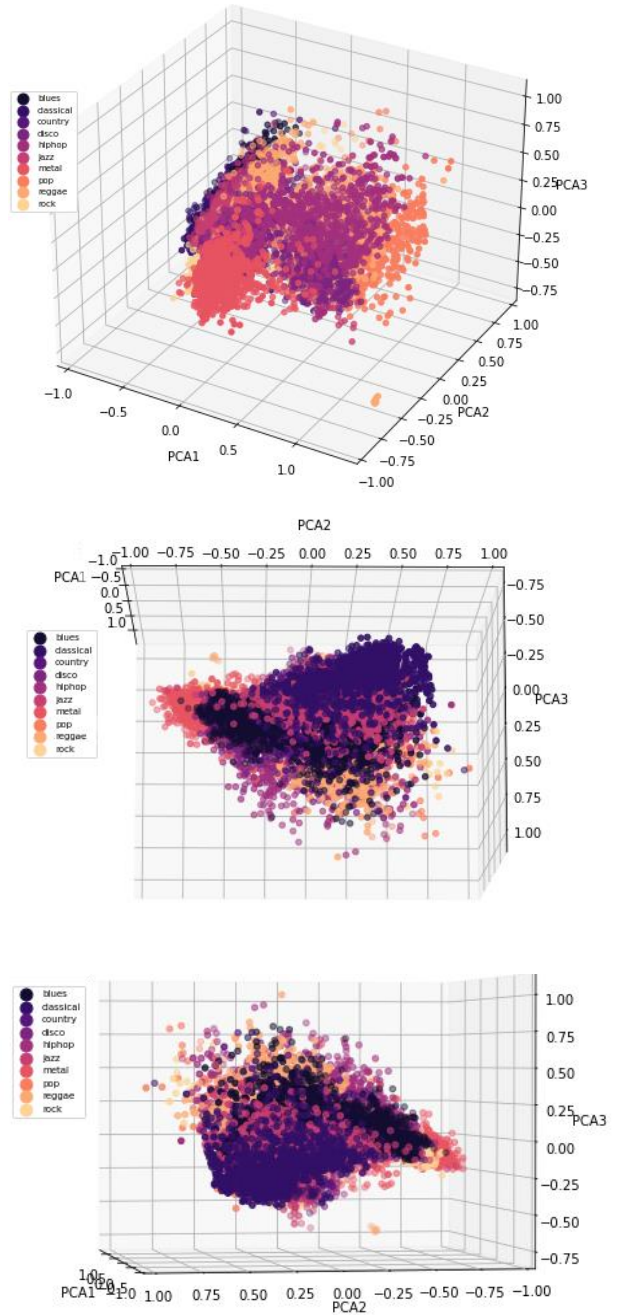
KKÖ yöntemini test etmek için, tüm etiketli veri kümesi üzerinde belirli oranlarda örneklerin sınıf etiketinin kaldırıldığı, etiket silme işlemi uygulandı. Dört farklı oranda (%86, %82, %74, %58) etiket silme işlemi yapılarak etiketli, etiketsiz ve test veri kümeleri elde edildi. Veri dengesizliğini önlemek için, tüm veri kümeleri sınıf dağılımı dengeli olacak şekilde bölündü. Test kümesi sabit (%10) olarak tutuldu.

1. Veri kümesinin test verisi dışında kalan kısmı (etiket silme olmadan)
2. Etiketsiz veri ayrıldıktan sonra kalan kısmı (daraltılmış etiketli veri),
3. Etiketsiz veriye KKÖ yöntemiyle sözde etiket atayarak oluşan veri kümeleri kullanılarak model eğitimi yapıldı. Kümeler örneklerin sıra numarası üzerinden ayrıldı. Böylece, her sınıflandırma algoritması için aynı eğitim ve test veri kümesi oluşumları kullanıldı.

2.2. Sınıflandırma Yöntemleri (Classification Methods)

Eğitim ve test çalışmaları *Naive Bayes*, *K-En Yakın Komşu*, *Karar Ağacı*, *Rastgele Orman* ve *XGBoost*, *Destek Vektör Makinesi*, *Lojistik Regresyon*, *Çok Katmanlı Yapay Sinir Ağı* yöntemleri kullanılarak yapılmıştır.

Naive Bayes, olasılık tabanlı bir yaklaşımla örüntü tanımlamada kullanılan bir denetimli makine öğrenmesi yöntemidir [33].



Şekil 3. Öznelik dağılım grafikleri (Feature scatter plots)

Naive Bayes yöntemi, tüm girdi özneliklerinin birbirinden bağımsız olduğu varsayımı üzerine kuruludur. $P(A)$, A olayının gerçekleşme olasılığı; $P(B)$, B olayının gerçekleşme olasılığı; $P(A|B)$, B olayı gerçekleşirken aynı zamanda A olayının gerçekleşme olasılığı; $P(B|A)$, A olayı gerçekleşirken aynı zamanda B olayının gerçekleşme olasılığıdır. Bu durumda, $P(A|B)$ olasılığı Eşitlik 3'teki gibi ifade edilir.

$$P(A | B) = \frac{P(B|A)*P(A)}{P(B)} \quad (3)$$

$x_1, x_2, x_3, \dots, x_n$ Şeklinde ifade edilebilecek bağımsız girdi öznelikleri için bağımlı hedef

özniteliğinin C_i sınıfı olma olasılığı Eşitlik 4'teki gibi ifade edilir.

$$P(C_i | X) = \frac{P(X | C_i) * P(C_i)}{P(X)}$$

$$(C_i | x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n | C_i) * P(C_i)}{P(x_1, \dots, x_n)} \quad (4)$$

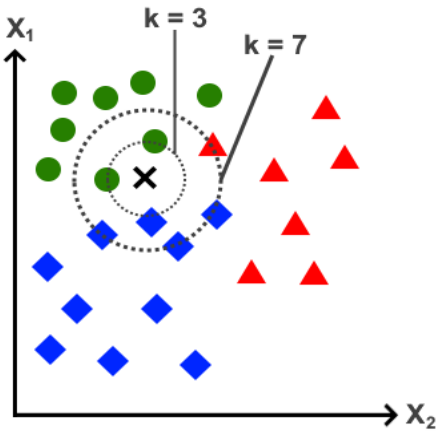
Sayısal türdeki bağımsız değişkenler için, girdi dizisinin ortalaması μ ve standart sapması σ olmak üzere bağımlı hedef özniteliğinin C_i sınıfı olma olasılığı *Gaussian Bayes* yöntemi ile hesaplanır ve Eşitlik 5'teki gibi ifade edilir.

$$\mu = \frac{1}{n} \sum_{j=1}^n x_j, \quad \sigma^2 = \frac{1}{n} \sum_{j=1}^n (x_j - \mu)^2$$

$$P(x_j | C_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right) \quad (5)$$

Her bir sınıf için, sınıfa ait olasılıkların çarpımı o sınıfın hedef değer olma olasılığını verir.

K-En Yakın Komşu (KNN - K-Nearest Neighbor), hiperuzayda bir noktaya ait hedef değerinin, kendisine en yakın noktalardan tahmin edildiği, sınıflandırma ve regresyon işlemleri için kullanılan bir denetimli öğrenme yöntemidir [34]. Bir denetimli öğrenme yöntemi olsa da, eğitim aşaması içermez. Şekil 4'te gösterildiği gibi hedef özniteliğinin değeri, çeşitli mesafe ölçüm yöntemlerine göre belirlenen k adet en yakın komşusuna bakılarak çoğunluk oyu (nokta sayısı) ya da ağırlıklı oy seçimi ile tespit edilir.



Şekil 4. K-en yakın komşu algoritması için en yakın komşuların belirlenmesi (Determination of nearest neighbors for the KNN algorithm)

Bağımsız girdi değişkenlerinin birbirleriyle orantısız olması sorun çıkaracağından, hesaplama öncesi normalizasyon yapılması gerekir. n Adet girdi özniteliğine sahip veri kümesindeki, i . öznitelik değerleri sırasıyla p_i , q_i ile temsil edilen

veri örnekleri P ve Q arasındaki mesafe Eşitlik 6'daki gibi hesaplanır.

$$d_r(P, Q) = \{\sum_{i=1}^n |p_i - q_i|^r\}^{\frac{1}{r}} \quad (6)$$

Uzaklık ölçüm yöntemleri, denklemden r yerine 1 koyulduğunda *Manhattan*, 2 koyulduğunda *Öklid*, 2'den büyük bir değer koyulduğunda *Minkowski* olarak adlandırılır. Kategorik örnekler için *Hamming mesafesi* uzaklık ölçüm yönteminden faydalanılır. Verideki eksikler/boşluklar ve aykırı değerler modelin çalışma başarısını düşürür; veri sayısı arttıkça performansı düşer.

Karar Ağacı (Decision Tree), sınıflandırma ve regresyon için kullanılan bir denetimli makine öğrenmesi yöntemidir [35]. Bağımsız girdi değişkenlerinin değer aralığı ya da kategorilerine göre koşullu (eğer ise, eğer değilse) olarak dallanmaların olduğu ikili ya da çoklu ağaç yapılarıyla temsil edilir. Homojen bir yapı oluşturmak için, ayrışmanın olacağı düğümlerin tespitinde *entropi*, *gini* ve *twoing* kriterleri kullanılır.

S veri örneklerinin bulunduğu küme, n hedef özniteliği değeri (sınıf) sayısı ve p_i küme içinde i . sınıfın bulunma olasılığı olduğu durumda, bu kümeye ait entropi $H(S)$ Eşitlik 7'deki gibi hesaplanır:

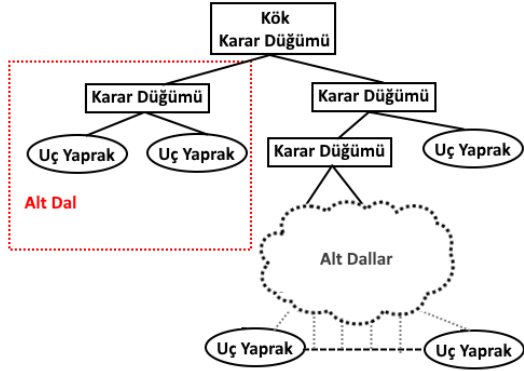
$$H(S) = -\sum_{i=1}^n p_i \log_2(p_i) \quad (7)$$

Karar ağaçları oluşturulurken ID3, C4.5, CART gibi algoritmalarından faydalanılır. ID3 sadece sayısal ya da sadece kategorik girdi öznitelikleri ile çalışır. C4.5 algoritması hem sayısal hem de kategorik girdi öznitelikleri ile çalışır. CART algoritması ikili ağaç yapısı oluşturur. CART algoritmasında *gini* ve *twoing* kriterleri kullanılır. ID3, C4.5 algoritmalarında *entropi* kriteri kullanılır ve ayırt edici özniteliklerin tespitinde "bilgi kazancı" ölçümünden yararlanır. X girdi özniteliği, S veri kümesi ve $S_1, S_2 \dots S_n$ alt kümeler olsun. $|S_i|$ i . kümenin, $|S|$ veri kümesinin tamamının eleman sayısını gösterecektir. X özniteliklerinin bilgi kazancı Eşitlik 8'deki gibi hesaplanır:

$$\text{Bilgi kazancı}(S, X) = H(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} H(S_i) \quad (8)$$

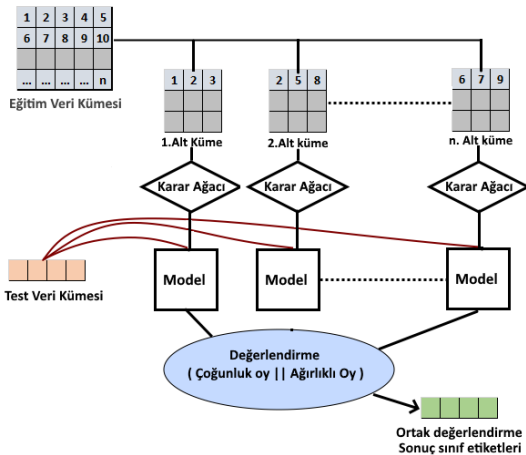
Şekil 5'te gösterildiği gibi karar ağacındaki örnekler ayrıştırma kriteri doğrultusunda alt dallara ayrılır. Kök düğüm, ayrıştırmanın başlayacağı kök özniteliklerini temsil eder ve belirlenen koşul doğrultusunda alt dallara ayrılarak gruplanır. Kök düğümün altında yer alan düğümler, küme homojen (sadece bir hedef öznitelik değerini içerecek) hale geldiyse "uç yaprak", yine dallanma varsa "karar

düğümü” olarak adlandırılır. Dallanma işlemi ağacın bütün dallarında yaprak düğümlere ulaşana ya da maksimum derinlik gibi sonlanma koşulu oluşuncaya kadar devam eder.



Şekil 5. Örnek karar ağacı (Example decision tree)

Rastgele Orman (Random Forest), sınıflandırma ve regresyon problemlerinde kullanılan, *Karar Ağacı* tabanlı bir denetimli öğrenme yöntemidir. Topluluk öğrenmesi (*Ensemble learning*) kategorisinde yer alır. *Karar Ağacı* gibi diğer algoritmalarla karşılaştırıldığında, eğitim verisine aşırı duyarlı olma eğilimindedirler [36]. Eğitim örneklerindeki değişiklikler, varyansın artmasına ve dolayısıyla aşırı öğrenmeye yol açabilir. Bu durumu önlemek için "torbalama" (*bagging, bootstrap aggregation*) adı verilen teknikler kullanılır. Bu yöntemde, eğitim kümesine ait örnekler rastgele şekilde eğitim gruplarına ayrılır. Bir örnek, bir ya da birden fazla grupta yer alabilir ya da hiçbir grubun içinde yer almaz. Ayırma işleminden sonra, eğitim gruplarıyla birden fazla karar ağacı birlikte eğitilir ve hata oranlarına göre çoğunluk oylaması veya ağırlıklı oy yöntemleri kullanılarak seçim yapılır [37]. Rastgele orman algoritmasının çalışma örneği Şekil 6'da gösterildiği gibidir.



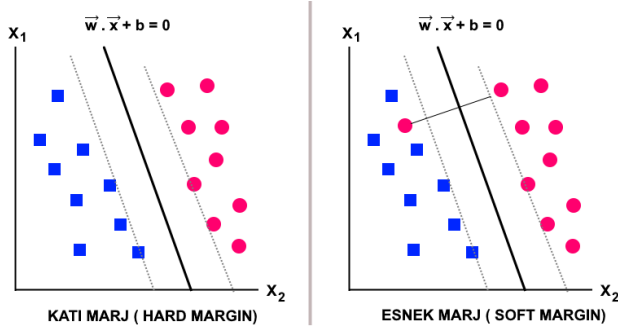
Şekil 6. Rastgele Orman algoritmasının örnek çalışma şekli (Example of how the random forest algorithm works)

XGBoost, sınıflandırma ve regresyon amacıyla kullanılan bir denetimli öğrenme algoritmasıdır. *Rastgele Orman* algoritması gibi topluluk öğrenmesi grubunda yer alır. İşleyiş olarak rastgele orman algoritmasına benzer. *Rastgele Orman* algoritmasında torbalama (*bagging*) yapılırken, XGBoost algoritmasında artırma (*boosting*) uygulanır.

Artırma yönteminde, birden fazla sınıflandırıcıdan gelen tahminler birleştirilerek sonuç elde edilir. Sınıflandırıcılar, sırayla bir öncekinin hatasına göre ağırlıklandırılarak (güçlü olanın ağırlığı artırılarak) sonuç modele ulaşılır.

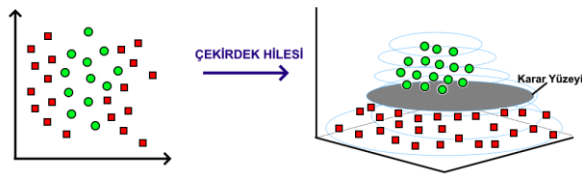
Alt eğitim kümelerindeki ağaçların dallarındaki hatalar hesaplanarak uç yapraklarda budama yapılır. Döngüsel olarak yapılan budama sonucunda her ağacın sadece belirli bir bölümü tahminlemede kullanılır. Böylece aşırı öğrenmenin önüne geçilir. Çalışma şekli bakımında XGBoost, *Gradyan Artırma Karar Ağacı Algoritmasının* optimize edilmiş halidir [38]. Hız, performans, uyarlanabilirlik, girdi çeşitliliği toleransı ve eksik veriyi işleyebilmesi özellikleri ile öne çıkar.

Destek Vektör Makineleri (Support Vector Machines), sınıflandırma amacıyla kullanılan bir denetimli makine öğrenmesi yöntemidir. Hem doğrusal olarak ayrılabilen hem de doğrusal olarak ayrılamayan veri kümelerinin sınıflandırılmasında kullanılır [39]. Doğrusal olarak ayrılabilen veri kümeleri için, hiperuzayda sınıfları ayıran uygun hiper düzlemi bulma esasına dayanır. Hiper düzlem tespit edilirken, her sınıfa ait veri örnekleri oluşturduğu sınır ile maksimum uzaklık sağlanmaya çalışılır. Sınıflar, sınırları belli olacak şekilde ayrılması durumunda, bu katı marj (*hard margin*) sınıflandırma olarak adlandırılır. Sınıflara ait veri örneklerinin bir kısmı ait oldukları marjın dışına taşmışsa bu esnek marj (*soft margin*) sınıflandırma olarak adlandırılır. Katı ve esnek marj sınıflandırmaya ait örnek gösterim Şekil 7'deki gibidir. Doğrusal olarak ayrılamayan veri kümelerinde ise, sınıfları ayırmak için çeşitli yöntemler kullanılır. Bunlardan biri çok terimli dönüştürmedir. Bu yöntemde boyut artırımı ile veri kümesi doğrusal ayrılabilir hale getirilir. Ancak bu yöntem hesaplama maliyetinde artışa, model performansında düşüşe ve aşırı öğrenmeye neden olur. İlgili sakıncaları aşmak adına, çekirdek hilesi (*kernel trick*) yöntemine başvurulur. Çekirdek hilesi yöntemi ile veri kümesi herhangi bir boyut artırma işlemi yapılmadan daha yüksek düzeyli bir uzaya eşleştirilir. Böylece esnek ve verimli bir şekilde ayrıştırma yapılır. Çekirdek hilesi yönteminde *Linear, Çok Terimli, Radyal Bazlı Fonksiyon (RBF) ve Sigmoid* gibi çekirdek fonksiyonları kullanılır.



Şekil 7. Katı marj ve esnek marj (Hard and soft margin)

Şekil 8’de iki adet girdi özneliğine sahip veri kümesinin RBF çekirdek hilesi ile dönüştürülmesi örneklenmiştir.



Şekil 8. RBF çekirdek hilesi (RBF Kernel trick)

Lojistik Regresyon (Logistic Regression), sınıflandırma amacıyla kullanılan bir denetimli makine öğrenmesi yöntemidir [40]. Bağımsız girdi değişkenlerinin ağırlık değerleri ile çarpılarak elde edilen toplamdan, olasılık değeri hesaplanır. Bu olasılık değeri ile hedef özneliği tahmin edilir. x_{ji} i. örnek için j. girdi özneliğinin değerini, b_j özneliğe ait ağırlığı ve e_j hatayı göstermek üzere i. örneğin öngörülen değeri \hat{Y}_i Eşitlik 9’da verildiği şekilde hesaplanır.

$$\hat{Y}_i = b_0x_{0i} + b_1x_{1i} + b_2x_{2i} \dots + b_kx_{ki} + e_i$$

$$= \sum_{j=0}^k b_jX_{ji},$$

$$\hat{Y}_i = b_0 + \sum_{j=1}^k b_jX_{ji} + e_i \quad (9)$$

İkili sınıflandırma ele alındığında, bir örneğe ait girdi öznelikleri X_i iken ilk sınıfın (hedef özneliği) olasılığı $P(X_i) = P(Y_i = 0, X_i)$ ikinci sınıfın olasılığı $P(Y_i = 1, X_i) = 1 - P(X_i)$ şeklinde bulunur. İki olasılığın birbirine bölümü ile iki sınıf olasılığı arasındaki oran tespit edilir. *Logit* fonksiyonu ile bu değer 0-1 arasındaki olasılık değerine çevrilir. Bu da Eşitlik 10’da gösterildiği gibi *sigmoid* fonksiyonuna karşılık gelmektedir.

$$Oran = \frac{P(X_i)}{1-P(X_i)}, \quad (10)$$

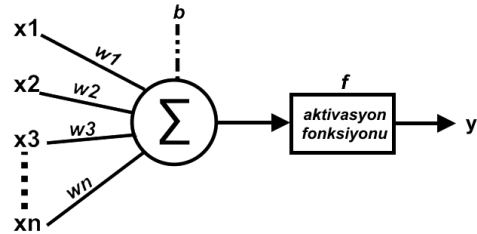
$$\text{logit}(Oran) = \log_e \left(\frac{p(X_i)}{1-p(X_i)} \right) \quad p(X_i) = \frac{e^{\hat{Y}_i}}{1+e^{\hat{Y}_i}}$$

İkili sınıflandırma için $p(X_i) \geq 0,5$ olması durumunda ilk sınıf (0) aksi halde ikinci sınıfa (1) ait hedef özneliği değeri elde edilir. Çok sınıflı lojistik regresyonda, sigmoid fonksiyonu yerine *softmax* fonksiyonu kullanılır. n Adet hedef özneliği değeri (sınıf etiketi) içeren bir veri kümesinde bir örneğe ait ϕ girdi öznelikleri için k . sınıf etiketi C_k ’nın olasılığı Eşitlik 11’deki gibi hesaplanır. Olasılığı en büyük sınıf, etiket hedef özneliği değeri olarak atanır.

$$p(C_k | \phi) = y_k(\phi) = \frac{\exp(\hat{Y}_k)}{\sum_{i=1}^N \exp(\hat{Y}_i)} \quad (11)$$

Girdi özneliklerine (bağımsız değişkenlere) bağlı ağırlıkların ve katsayıların hesaplanmasında *En Büyük Olabilirlik tahmini (MLE - Maximum Likelihood Estimator)*, *Yeniden Ağırlıklandırılmış En Küçük Kareler (RILS - Iteratively Reweighted Least Squares)* ve *Gradyan İniş (Gradient Descent)* yöntemlerinden faydalanılır.

Çok Katmanlı Yapay Sinir Ağı/ÇKYSA (Multilayer Neural Network), biyolojik nöronların mekanik yapısının taklit edilmesiyle ortaya çıkan makine öğrenmesi yöntemidir [41]. Hata toleransının yüksek olması, doğrusal olmayan problemler için uygun olması, uyarlanabilir olması gibi özelliklerden dolayı yaygın olarak kullanılır. Sinir ağı katmanları, yan yana gelen algılayıcılardan (*perceptron*) oluşur. Sinir ağlarına ait algılayıcılar Şekil 9’daki gibi çalışır.



Şekil 9. McCulloch–Pitt yapay nöron modeli (McCulloch–Pitt Model) [42]

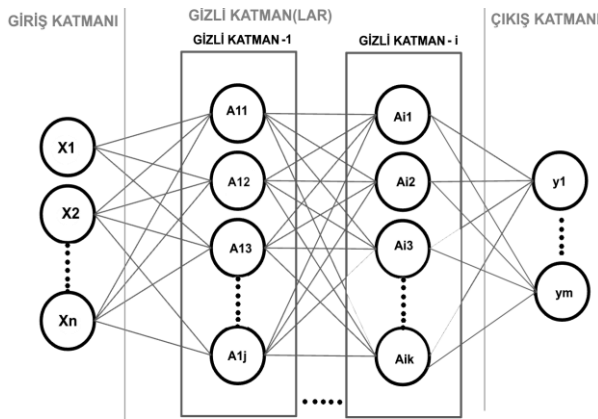
Eşitlik 12’de verildiği gibi, algılayıcı girdileriyle ($x_1, x_2, x_3 \dots x_n$) eğitim aşamasında hesaplanan ağırlık değerleri ($w_1, w_2, w_3 \dots w_n$) çarpılıp, bir taban değeri b ile toplandıktan sonra aktivasyon fonksiyonu f kullanılarak hedef değeri elde edilir.

$$f(b + \sum_{i=1}^n x_i w_i) \quad (12)$$

Problemin niteliğine göre *sigmoid*, *Tanh*, *Relu*, *LeakRelu*, *Softmax*, *Swing* vb. aktivasyon fonksiyonları kullanılır. Çok katmanlı sinir ağı Şekil 10’da gösterildiği şekilde, bir *girdi katmanı (input*

layer), bir ya da birden fazla gizli katman (hidden layer) ve çıktı katmanından (output layer) oluşur.

Birbirinden bağımsız birden fazla algılayıcının yan yana gelmesiyle katmanlar oluşur. Katmandaki algılayıcıların çıktılarının diğer katmanlardaki algılayıcıların girdisi olacak şekilde daha fazla katmanın bir araya gelmesiyle çok katmanlı yapay sinir ağları meydana gelir. Sinir ağlarında, gerçek ve tahmin edilen hedef özniteliği değeri karşılaştırılarak geri yayılım (back propagation) algoritması ile ağırlık değerleri güncellenir. İleri yayılım (forward propagation) ile hesaplanan yeni ağırlık değerleri ile tekrar tahminleme yapılır ve bu döngüler halinde tekrarlanır.



Şekil 10. Çok katmanlı sinir ağı mimarisi (Multi layer neural network architecture)

2.3. Performans Değerlendirme Ölçütleri (Performance Evaluation Metrics)

Karışıklık Matrisi (Confusion Matrix): Karışıklık matrisi, bir girdi örneğinin ait olduğu sınıf etiketi ile sınıflandırıcının tahmin ettiği sınıf etiketinin karşılaştırılmasını sağlayan tablodur. Şekil 11’de ikili sınıflandırmaya ait örnek karışıklık Matrisi verilmiştir.

		TAHMİN	
		Pozitif	Negatif
GERÇEK	Pozitif	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Şekil 11. Karışıklık matrisi (Confusion Matrix)

Doğruluk (Accuracy): Doğru tahminlerin toplam miktarının, kümedeki örnek sayısına bölümüyle elde edilir. Özellikle dengesiz kümelerde başarıyı ölçmek için yeterli olmayacağından diğer ölçütlerle beraber kullanılır. Eşitlik 13’teki gibi ifade edilir.

$$\text{Doğruluk} = \frac{DP + DN}{DP + YP + DN + YN} \quad (13)$$

Kesinlik (Precision): Pozitif sınıf için, doğru tahmin edilen örnek sayısının, doğru ve yanlış pozitif tahmin örnekleri toplamına bölümünden oluşur. Pozitif öngörü oranıdır. Eşitlik 14’teki gibi ifade edilir.

$$\text{Kesinlik} = \frac{DP}{DP + YP} \quad (14)$$

Duyarlılık (Sensitivity / Recall): Pozitif sınıf için doğru tahminlerin, gerçek pozitif örneklerin sayısına bölümünden oluşur. Pozitif gerçek değerler için doğruluk oranıdır. Eşitlik 15’teki gibi ifade edilir.

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \quad (15)$$

Özgünlük (Specify): Negatif sınıf için, doğru tahminlerin, gerçek negatif örneklerin sayısına bölümünden oluşur. Negatif gerçek değerler için doğruluk oranıdır. Eşitlik 16’daki gibi ifade edilir.

$$\text{Özgünlük} = \frac{DN}{DN + YP} \quad (16)$$

F1-Skor (F1-Score): Duyarlılık ve özgünlük değerleri dikkate alarak model başarısını ölçmekte kullanılır. Eşitlik 17’deki gibi ifade edilir.

$$F_1 - \text{skor} = 2 \times \frac{\text{Duyarlılık} \times \text{Özgünlük}}{\text{Duyarlılık} + \text{Özgünlük}} \quad (17)$$

3. BULGULAR (RESULTS)

Deneysel çalışma için dört eğitim ve test kümesi oluşturuldu.

Etiket kaldırma oranı (tüm küme üzerinden sırasıyla %86, %82, %74, %58) belirlendi. %10’lu kısım test veri kümesi olarak belirlendikten sonra sırasıyla kalan %4, %8, %16, %32’lik bölümü kısmi eğitim kümesi olarak kullanıldı.

1. %4, %8, %16, %32’lik kısmi eğitim kümesi,
2. Test kümesi dışında kalan %90’lık tümü etiketli eğitim kümesi,
3. Kısmi eğitim kümeleri ile sırasıyla %86, %82, %74, %58’lik etiketsiz veriye sözde etiket atanarak oluşan kümeler ile eğitim ve test işlemleri yürütüldü.

Deneysel çalışmalarda, sözde etiket atama işlemi için güvenilirlik (olasılık eşik değeri) 0,98 olarak seçildi. K-En Yakın Komşu sınıflandırıcısı için $k=3$ değerinde dirsek noktası olduğundan, bakılacak komşu sayısı 3 olarak belirlendi. Yapay Sinir Ağı modeli, düğüm sayısı sırasıyla 512, 256, 128, 64

olan 4 gizli katmana sahip olacak şekilde oluşturuldu. Gizli katmanlarda *RELU* aktivasyon fonksiyonu kullanıldı. Sınıf sayısı kadar (10) düğümüne sahip çıkış katmanında aktivasyon fonksiyonu olarak *softmax* kullanıldı. Aşırı öğrenmeyi engellemek adına gizli katmanların çıkışında 0,25 oranında seyreltme uygulandı. Öğrenme oranı 0,0001 olacak şekilde *RMSProp* optimizasyon fonksiyonu kullanıldı. Lojistik regresyon modeli için maksimum döngü sayısı 1000, $\gamma = 0,0001$, çekirdek *linear* olacak şekilde parametre düzenlemesi yapıldı. *XGBoost* modelinde eğitilecek karar ağacı sayısı 100 olarak belirlendi. Diğer sınıflandırıcılar için varsayılan parametreler üzerinden yürütüm gerçekleştirildi.

Deneyisel çalışmalarda, Grafik İşlem Birimi içermeyen, toplam 20 çekirdekli 2 adet Intel *Xeon SR650* raf sunucu kullanıldı. Geliştirme ortamı olarak *Anaconda*, programlama dili olarak *Python* ve bu dile ait *numpy*, *pandas*, *matplotlib*, *sklearn*, *keras*, *semilearn* kütüphaneleri kullanıldı.

Deneyisel çalışmaya ait sonuçlar Tablo 2, 3, 4, 5 ve 6'da verilmiştir. Tablo 2'de tüm sınıflandırıcılar için tamamı etiketli veri kümesi üzerinden yapılan ölçüm değerleri yer almaktadır. En düşük etiketli veri oranından en yükseğe doğru Tablo 3, 4, 5 ve 6'da kısmi etiketli ve sözde etiket atama ile oluşan veri kümelerine ait ölçümler göstermektedir. Tabloların ilk sütununda kullanılan sınıflandırıcı bilgisi, ikinci sütununda kullanılan veri kümesi (VK) yer almaktadır. **VK** sütununda 1 değeri kısıtlı, 2 değeri sözde etiket atama işlemi sonrası, 3 değeri tümü etiketlenmiş eğitim kümesi ile yapılan testleri ifade eder. Sonuç etiketi bazında **F₁-Skor** değeri, kümenin tümüne ait doğruluk değeri (**Doğruluk**) ölçütlerini içeren sütunlar yer almaktadır. Ayrıca, tablolarda, sözde etiket atama işleminde atama yapılan örnek (**EA**), doğru atama yapılan örnek sayısı (**DEA**) ve bunların oranını (**DEA/EA**) gösteren sütunlar yer almaktadır. Sözde etiket atama işleminin kaç döngüde tamamlandığı **Tekrar** ve işlem süresi bilgisi **Süre** sütunlarında gösterilmiştir.

- Eğitim kümesi boyutu arttıkça KNN, ÇKYSA ve XGBoost modellerinde doğru sözde etiket atanan örnek sayısı oranı artmıştır. KNN için Tablo 3-6'da gösterildiği gibi DEA/EA oranı sırasıyla %72, %86, %90, %96'dır. Benzer şekilde ÇKYSA için DEA/EA oranı sırasıyla %75, %84, %89, %96; XGBoost için %70, %81, %92, %98'dir. Diğer modellerde belirgin bir artış olmamıştır ya da model doğruluk oranı düşük olduğu için DEA/EA oranındaki artış model performansına yansımamıştır.

- 1. ve 3. veri kümeleri karşılaştırıldığında, beklenildiği üzere, eğitim kümesi boyutundaki artışa koşut olarak işlem süreleri de artmıştır. Sözde etiket atama işleminin yapıldığı 2. veri kümesi için, işlem süresi tekrar sayısı ve eğitim veri kümesinin boyutuna bağlı olarak değişmiştir.
- Kullanılan eğitim kümesi boyutu büyüdükçe doğruluk ve sınıf bazında F₁-skor ölçütlerinde iyileşme olmuştur.
- Eğitim kümesi boyutu büyüdükçe, sözde etiket atama tekrar sayısında genel olarak azalma olmuştur. Örneğin ÇKYSA modeli için Tablo 3'te verilen yapılandırmada sözde etiket atama işlemi 4 tekrarda tamamlanırken, Tablo 4, 5, 6'da yer alan yapılandırmalarda sırasıyla 3, 3 ve 1 tekrarda işlem sonlanmıştır.

Tablo 2, 3, 4, 5 ve 6'da yer alan veriler toplu olarak değerlendirildiğinde aşağıdaki bulgulara ulaşılmıştır:

- KKÖ'nün, sınıflandırma başarısı düşük olan Naive Bayes, Lojistik Regresyon ve Karar Ağacı modelleriyle beraber kullanıldığında başarıyı düşürdüğü görülmüştür. Sınıflandırıcı başarısının sözde etiket atama işlemine de yansıdığı düşünüldüğünde, bu beklenen bir sonuçtur.
- Karar ağacı tabanlı, topluluk öğrenmesi yöntemi Rastgele Orman ve XGBoost modellerinde, sözde etiket atama başarısı yüksek olsa dahi KKÖ'nün başarı artışı sağlamadığı ya da düşürdüğü görülmüştür.
- Destek Vektör Makinesi modelinde, sözde etiket atama işlemi yüksek oranda başarılı olsa da, sözde etiket atanan örnek sayısı azdır. Bu nedenle KKÖ yönteminin kullanılması başarıyı etkilememiştir.
- KNN modeli için KKÖ doğruluk oranı yönünden 0,02-0,04 arası artış sağlamıştır.
- ÇKYSA modeli için KKÖ 0,05-0,09 arasında istikrarlı bir doğruluk oranı artışı sağlamıştır.

Ek Yöntem: KKÖ'nün en çok katkısını sağladığı KNN ve ÇYSSA sınıflandırıcılarda aşağıdaki adımları içeren destekleyici bir yöntem uygulanmıştır.

- 1- Eşik değerinin üstünde olasılığa sahip etiketlerin sınıf bazında toplamını sayısını bul.
- 2- En düşük miktara sahip sınıfın belirli bir yüzdesini (bu çalışmada %80 alınmıştır) **etiket atama oranı** olarak belirle.

3- Her sınıf için, eşik değerin üstündeki örnekler için, **etiket atama oranı** kadar olan kısmına etiket ata.

Bu yöntemle, etiket atama sırasında sınıflar arasındaki dengesiz dağılımının önüne geçilmesi amaçlanmıştır. Tablo 3, 4, 5 ve 6'da, anılan iki sınıflandırıcı için 2* şeklinde verilen satırlarda ek yöntemle ait sonuçlar görülmektedir. Buna göre, ek yöntem doğru etiket atama oranını yükseltmiş, 0,01-0,02 arasında ek bir doğruluk artışı sağlamıştır. Çok büyük bir artışı ifade etmese de bu, destekleyici yöntemlerin sağlayabileceği katkının potansiyelini göstermesi yönünden anlamlıdır.

KKÖ gibi sarıcı yöntemler ele alındığında, bu yöntemler için temel dayanak aynı sınıfa ait örneklerin belirli bir bölgede toplanmasıdır. Ancak mevcut çalışmadaki gibi veri kümesinin geniş yayılıma sahip olması, KKÖ yönteminin performansını olumsuz etkilemiştir. Şekil 3'teki dağılım grafiğinde de görüleceği üzere, kimi müzik türleri belirli bölgelerde yoğunlaşırken, rock gibi müzik türleri için örnekler geniş bir bölgeye dağılmıştır. Bu durum, ilgili türler için doğruluk oranı ve F1-skor değerinin düşmesine sebep olmuştur. Belirli bir bölgede yoğunlaşan *classic* ve *metal* müzik türleri için yüksek F1-skor değeri elde edilmiştir. Bunun sonucunda, sözde etiket atama işleminde, sadece belirli bir bölgede toplanan örneklerle sahip müzik türlerine yüksek oranda ve başarılı bir şekilde sözde etiket atanırken, diğer müzik türleri için etiket atama işleminin başarısı düşmüştür. Başka bir ifadeyle, yüksek öbeklenme olan müzik türleri için ölçüm değerlerinde iyileşme yaşanırken; diğer müzik türlerinde model başarısı düşmüş ve yanlış etiketleme oranı artmıştır.

Son olarak, mevcut çalışmada elde edilen sonuçlar, Wang ve diğerleri tarafından yapılan çalışmadaki [25] sonuçlarla karşılaştırılmıştır (Tablo 7). Tabloda da görüleceği üzere, referans çalışmada sözde etiket atama (*Pseudo-labelling*) yöntemi hata oranını arttırmıştır. Öte yandan *FixMatch*, *FlexMatch* algoritmaları hata oranını ortalama %10 (bağıl olarak yaklaşık %30) azaltmıştır. Mevcut çalışmada, aynı yapılandırma üzerinde yapılan deneylerde KKÖ destekli ÇKYSA modeli ile hata oranında %6 (bağıl olarak yaklaşık %14) iyileşme sağlanmıştır.

4. SONUÇ (CONCLUSION)

Çalışma kapsamında, KKÖ'nün GTZAN veri kümesi üzerinden sınıflandırma performansına etkisi değerlendirilmiştir. KKÖ, her ne kadar umut vaat eden bir yarı denetimli öğrenme yöntemi olsa da, her durumda istenen katkıyı sağlamamaktadır. Kullanılan veri kümesinin niteliğine göre kimi

durumlarda başarı artışı sağlarken, kimi durumlarda etkisiz kalmakta ya da başarıyı düşürmektedir.

Örneğin, Li ve Zhou'nun çalışmasında [17], veri kümesindeki dağılım ve gürültü miktarına bağlı olarak, KKÖ herhangi bir ek teknik ile desteklenmediğinde performans artışı yerine düşüşe sebep olabileceği görülmektedir. Bu nedenle, KKÖ SETRED adı verilen algoritma ile desteklenmiştir.

Bir başka çalışmada [24] önerilen gürültü giderme, çalışma kapsamında işlenen veri kümelerinde başarılı olmuştur. Ancak, yöntem, sınıfların geniş dağılıma sahip olduğu bu çalışmada kullanılan benzer veri kümelerinde, dağılık örneklerin gürültü olarak algılanmasına sebep olmaktadır. Bu da, sınıflandırma başarısını düşürmektedir. Sonuç olarak, KKÖ vb. sarıcı yöntemlerin başarılı sonuç üretebilmesi için:

- Veri kümesinin yapısına uygun, yeterli performansa sahip bir model seçilmelidir.
- Kısıtlı veri kümesinin tüm hedef sınıfları için yeterli örneğe sahip olması ve sınıfların düzenli bir dağılım göstermesi gerekmektedir.
- Veri kümesinin yapısına göre, gerekli durumlarda yöntem, gürültü filtreleme ya da etiket ve model bazında düzenleme gibi ek tekniklerle desteklenmelidir.

KKÖ'nün yalın halde kullanımı yerine, veri kümesinin karakteristiğine bağlı olarak, yukarıda bahsedilenler gibi ek tekniklerle desteklenmesinin model başarısını artıracığı düşünülmektedir.

ETİK STANDARTLARIN BEYANI (DECLARATION OF ETHICAL STANDARDS)

Bu makalenin yazarı çalışmalarında kullandıkları materyal ve yöntemlerin etik kurul izni ve/veya yasal-özel bir izin gerektirmediğini beyan ederler.

The author of this article declares that the materials and methods they use in their work do not require ethical committee approval and/or legal-specific permission.

YAZARLARIN KATKILARI (AUTHORS' CONTRIBUTIONS)

Alp Kaan TURAN: Analiz, yazılım uygulaması, sonuçların değerlendirilmesi ve makalenin yazım işlerini yürütmüştür.

He performed the analysis, software implementation, analysis of the results and writing the paper.

Hüseyin POLAT: Gözden geçirme, yönlendirme ve makalenin yazımı konularında katkı sağlamıştır.

He contributed to the review, supervision and writing of the paper.

ÇIKAR ÇATIŞMASI (CONFLICT OF INTEREST)

Bu çalışmada herhangi bir çıkar çatışması yoktur.

There is no conflict of interest in this study.

Tablo 2. %90 etiketli küme (90% labeled dataset)

Sınıflandırıcı	VK	F1-Skor											Doğruluk	Süre
		blues	classic	country	disco	hip hop	jazz	metal	pop	reggae	rock			
K-En Yakın Komşu	3	0,94	0,93	0,88	0,91	0,92	0,91	0,96	0,93	0,94	0,87	0,92	< 1sn	
Ç,K, Yapay Sınır Ağı	3	0,93	0,90	0,80	0,89	0,86	0,85	0,93	0,91	0,87	0,79	0,87	04:39	
Destek Vektör Makinesi	3	0,58	0,88	0,38	0,54	0,54	0,72	0,79	0,74	0,58	0,43	0,63	00:21	
XGBoost	3	0,90	0,94	0,88	0,89	0,90	0,85	0,94	0,90	0,89	0,88	0,90	00:09	
Rastgele Orman	3	0,92	0,94	0,85	0,84	0,87	0,88	0,91	0,90	0,89	0,80	0,88	01:50	
Naive Bayes	3	0,39	0,79	0,47	0,38	0,40	0,53	0,60	0,65	0,47	0,26	0,51	< 1sn	
Karar Ağacı	3	0,67	0,83	0,52	0,55	0,64	0,71	0,74	0,67	0,62	0,51	0,65	00:01	
Lojistik Regresyon	3	0,66	0,89	0,54	0,67	0,63	0,75	0,83	0,79	0,62	0,50	0,69	00:02	

Tablo 3. %4 etiketli, %86 etiketsiz veri kümesi (4% labeled, 86% unlabeled dataset)

Sınıflandırıcı	VK	F1-Skor											Doğruluk	EA	DEA	DEA/EA	Tekrar	Süre
		blues	classic	country	disco	hip hop	jazz	metal	pop	reggae	rock							
K-En Yakın Komşu	1	0,51	0,80	0,51	0,48	0,51	0,61	0,71	0,64	0,59	0,42	0,57						<1sn
	2	0,57	0,79	0,52	0,46	0,59	0,65	0,75	0,70	0,60	0,43	0,60	5070	3626	0,72	4	<1sn	
	2*	0,57	0,86	0,47	0,58	0,50	0,71	0,73	0,69	0,64	0,37	0,62	4402	3569	0,81	10	00:05	
Ç,K, Yapay Sınır Ağı	1	0,56	0,82	0,51	0,34	0,44	0,66	0,65	0,72	0,50	0,32	0,56						00:18
	2	0,59	0,80	0,52	0,53	0,62	0,68	0,75	0,72	0,57	0,47	0,62	5238	3953	0,75	4	06:01	
	2*	0,59	0,89	0,58	0,49	0,61	0,66	0,75	0,74	0,64	0,38	0,64	4357	3751	0,86	7	08:38	
Destek Vektör Makinesi	1	0,34	0,74	0,00	0,06	0,07	0,39	0,52	0,56	0,39	0,00	0,40						<1sn
	2	0,34	0,74	0,00	0,06	0,07	0,39	0,52	0,56	0,39	0,00	0,40					1	<1sn
	1	0,59	0,83	0,53	0,50	0,59	0,64	0,70	0,72	0,58	0,35	0,61						00:01
XGBoost	2	0,48	0,74	0,41	0,49	0,51	0,59	0,66	0,62	0,53	0,28	0,52	4320	3022	0,70	29	01:54	
	1	0,59	0,89	0,58	0,49	0,61	0,66	0,75	0,74	0,64	0,38	0,64						00:04
	2	0,59	0,89	0,54	0,49	0,60	0,69	0,74	0,74	0,62	0,34	0,63	2	2	1,00	3	00:22	
Rastgele Orman	1	0,40	0,73	0,47	0,29	0,41	0,48	0,62	0,59	0,48	0,23	0,47						<1sn
	2	0,31	0,64	0,48	0,33	0,38	0,43	0,61	0,51	0,42	0,25	0,43	4910	2850	0,58	2	<1sn	
	1	0,24	0,76	0,18	0,23	0,36	0,40	0,49	0,53	0,38	0,24	0,38						<1sn
Karar Ağacı	2	0,23	0,68	0,20	0,19	0,32	0,38	0,49	0,55	0,36	0,24	0,37	6250	2883	0,46	1	<1sn	
	1	0,47	0,81	0,28	0,26	0,49	0,58	0,60	0,66	0,53	0,19	0,52						<1sn
	2	0,47	0,81	0,28	0,26	0,49	0,58	0,60	0,66	0,53	0,19	0,52						<1sn
Lojistik Regresyon	1	0,47	0,81	0,28	0,26	0,49	0,58	0,60	0,66	0,53	0,19	0,52						<1sn
	2	0,47	0,81	0,28	0,26	0,49	0,58	0,60	0,66	0,53	0,19	0,52						<1sn
	1	0,47	0,81	0,28	0,26	0,49	0,58	0,60	0,66	0,53	0,19	0,52						<1sn

Tablo 4. %8 etiketli, %82 etiketsiz veri kümesi (8% labeled, 82% unlabeled dataset)

Sınıflandırıcı	VK	F1-Skor											Doğruluk	EA	DEA	DEA/EA	Tekrar	Süre
		Blues	classic	country	disco	hip hop	jazz	metal	pop	reggae	rock							
K-En Yakın Komşu	1	0,67	0,87	0,59	0,62	0,66	0,71	0,77	0,78	0,66	0,55	0,69						< 1sn
	2	0,76	0,89	0,65	0,68	0,66	0,70	0,80	0,85	0,63	0,58	0,72	4840	4151	0,86	3	< 1sn	
	2*	0,78	0,88	0,60	0,67	0,66	0,74	0,84	0,82	0,62	0,59	0,73	4472	4104	0,92	12	00:05	
Ç,K, Yapay Sınır Ağı	1	0,60	0,86	0,57	0,56	0,60	0,68	0,82	0,72	0,61	0,36	0,65						00:38
	2	0,69	0,94	0,62	0,62	0,67	0,80	0,86	0,77	0,66	0,61	0,72	5560	4645	0,84	3	03:18	
	2*	0,74	0,87	0,65	0,68	0,69	0,76	0,85	0,82	0,71	0,53	0,73	4890	4456	0,91	4	04:19	
Destek Vektör Makinesi	1	0,30	0,77	0,23	0,19	0,43	0,28	0,58	0,55	0,53	0,07	0,44						< 1sn
	2	0,30	0,77	0,23	0,19	0,43	0,28	0,58	0,55	0,53	0,07	0,44	15	14	0,93	6	00:04	
	1	0,65	0,88	0,57	0,64	0,72	0,69	0,86	0,74	0,68	0,54	0,70						00:02
XGBoost	2	0,57	0,84	0,47	0,64	0,70	0,68	0,75	0,70	0,61	0,45	0,64	3975	3209	0,81	40	03:29	
	1	0,67	0,87	0,57	0,61	0,71	0,74	0,83	0,76	0,62	0,55	0,70						00:08
	2	0,67	0,87	0,59	0,66	0,69	0,74	0,83	0,75	0,61	0,55	0,70	97	97	1,00	21	03:36	
Rastgele Orman	1	0,28	0,82	0,45	0,29	0,53	0,53	0,58	0,63	0,53	0,26	0,50						< 1sn
	2	0,24	0,82	0,39	0,25	0,52	0,50	0,60	0,52	0,47	0,27	0,45	4210	2502	0,59	4	< 1sn	
	1	0,37	0,81	0,25	0,34	0,44	0,57	0,61	0,54	0,36	0,34	0,46						< 1sn
Karar Ağacı	2	0,37	0,82	0,21	0,34	0,45	0,55	0,62	0,57	0,36	0,32	0,46	7050	3562	0,51	1	< 1sn	
	1	0,46	0,87	0,39	0,37	0,56	0,58	0,72	0,66	0,56	0,35	0,56						< 1sn
	2	0,46	0,87	0,39	0,37	0,56	0,58	0,72	0,66	0,56	0,35	0,56	1	1	1,00	2	< 1sn	

Tablo 5. %16 etiketli, %74 etiketsiz veri kümesi (16% labeled, 74% unlabeled dataset)

Sınıflandırıcı	VK	F ₁ -Skor											Doğruluk	EA	DEA	DEA/EA	Tekrar	Süre
		blues	classic	country	disco	hip hop	jazz	metal	pop	reggae	rock							
K-En Yakın Komşu	1	0,81	0,85	0,68	0,70	0,71	0,76	0,87	0,81	0,73	0,62	0,75						< 1sn
	2	0,85	0,87	0,68	0,71	0,73	0,79	0,89	0,83	0,77	0,66	0,79	4210	3807	0,90	3	< 1sn	
	2*	0,85	0,89	0,72	0,71	0,76	0,81	0,87	0,85	0,81	0,71	0,80	3980	3873	0,97	4	00:01	
Ç.K, Yapay Sınır Ağı	1	0,75	0,87	0,58	0,66	0,63	0,75	0,84	0,84	0,70	0,49	0,72						01:01
	2	0,79	0,89	0,75	0,77	0,73	0,78	0,80	0,83	0,76	0,65	0,77	4300	3831	0,89	3	03:18	
	2*	0,80	0,90	0,74	0,77	0,70	0,81	0,85	0,87	0,73	0,70	0,79	3880	3766	0,97	2	03:10	
Destek Vektör Makinesi	1	0,38	0,79	0,31	0,25	0,31	0,53	0,60	0,60	0,56	0,26	0,49						00:01
	2	0,38	0,79	0,31	0,25	0,31	0,53	0,60	0,60	0,56	0,26	0,49	145	139	0,96	10	00:19	
	1	0,79	0,90	0,71	0,71	0,70	0,77	0,86	0,80	0,68	0,65	0,76						00:05
XGBoost	2	0,79	0,90	0,68	0,71	0,72	0,78	0,86	0,77	0,73	0,61	0,75	3730	3414	0,92	26	02:51	
Rastgele Orman	1	0,74	0,87	0,65	0,68	0,69	0,76	0,85	0,82	0,71	0,53	0,73						00:16
	2	0,77	0,89	0,63	0,69	0,69	0,77	0,85	0,82	0,74	0,52	0,74	92	91	0,99	17	05:28	
	1	0,33	0,81	0,41	0,47	0,36	0,56	0,61	0,70	0,48	0,26	0,52						< 1sn
Naive Bayes	2	0,29	0,82	0,38	0,33	0,41	0,48	0,63	0,62	0,43	0,28	0,46	3700	2288	0,62	4	< 1sn	
	1	0,47	0,72	0,33	0,38	0,50	0,54	0,64	0,59	0,49	0,31	0,49						< 1sn
Karar Ağacı	2	0,46	0,75	0,35	0,38	0,47	0,49	0,62	0,60	0,45	0,32	0,49	6480	3415	0,53	1	< 1sn	
Lojistik Regresyon	1	0,57	0,86	0,47	0,58	0,50	0,71	0,73	0,69	0,64	0,37	0,62						< 1sn
	2	0,57	0,86	0,47	0,58	0,50	0,71	0,73	0,69	0,64	0,37	0,62	4	4	1,00	2	00:01	

Tablo 6. %32 etiketli, %58 etiketsiz veri kümesi (32% labeled, 58% unlabeled dataset)

Sınıflandırıcı	VK	F ₁ -Skor											Doğruluk	EA	DEA	DEA/EA	Tekrar	Süre
		blues	classic	country	disco	hip hop	jazz	metal	pop	reggae	rock							
K-En Yakın Komşu	1	0,88	0,94	0,77	0,80	0,83	0,81	0,95	0,83	0,88	0,70	0,84						< 1sn
	2	0,88	0,93	0,79	0,80	0,88	0,86	0,94	0,86	0,89	0,73	0,86	3350	3231	0,96	2	< 1sn	
	2*	0,86	0,92	0,85	0,87	0,88	0,87	0,92	0,87	0,87	0,77	0,87	3270	3253	0,99	3	00:01	
Ç.K, Yapay Sınır Ağı	1	0,73	0,82	0,73	0,69	0,77	0,77	0,87	0,83	0,75	0,54	0,75						03:42
	2	0,85	0,93	0,80	0,80	0,84	0,87	0,89	0,87	0,81	0,76	0,84	3300	3158	0,96	1	06:48	
	2*	0,87	0,94	0,79	0,86	0,90	0,84	0,92	0,89	0,89	0,75	0,86	3900	3783	0,97	2	07:09	
Destek Vektör Makinesi	1	0,52	0,88	0,38	0,28	0,49	0,65	0,67	0,64	0,62	0,27	0,56						00:03
	2	0,52	0,88	0,38	0,28	0,49	0,65	0,67	0,64	0,62	0,27	0,56	239	232	0,97	9	00:47	
XGBoost	1	0,81	0,95	0,80	0,78	0,85	0,87	0,92	0,86	0,86	0,70	0,84						00:08
	2	0,82	0,95	0,78	0,77	0,86	0,88	0,92	0,86	0,85	0,74	0,84	2925	2865	0,98	34	04:45	
Rastgele Orman	1	0,81	0,94	0,75	0,75	0,86	0,83	0,88	0,85	0,81	0,65	0,82						00:34
	2	0,80	0,94	0,75	0,75	0,85	0,85	0,88	0,86	0,79	0,68	0,82	108	108	1,00	22	13:56	
Naive Bayes	1	0,37	0,88	0,48	0,41	0,43	0,61	0,62	0,65	0,48	0,34	0,54						< 1sn
	2	0,32	0,87	0,41	0,34	0,41	0,53	0,61	0,59	0,40	0,26	0,48	2899	1737	0,60	22	00:01	
Karar Ağacı	1	0,55	0,81	0,52	0,54	0,59	0,61	0,68	0,71	0,61	0,32	0,59						< 1sn
	2	0,48	0,84	0,45	0,53	0,57	0,68	0,75	0,68	0,60	0,39	0,59	5460	3129	0,57	1	00:01	
Lojistik Regresyon	1	0,56	0,89	0,53	0,55	0,60	0,68	0,77	0,71	0,67	0,42	0,64						< 1sn
	2	0,56	0,89	0,53	0,55	0,60	0,68	0,77	0,71	0,67	0,42	0,64	38	38	1,00	5	00:04	

Tablo 7. Yöntem karşılaştırma tablosu (Method comparison table)

Yöntem	Referans	Hata Oranı	Doğruluk
Tamamı Etiketli	[25]	5,98 ± 0,32	94,02 ± 0,32
Kısmi Etiketli (%4)		31,53 ± 0,52	68,47 ± 0,52
Pseudo-Labeling		33,93 ± 0,69	66,07 ± 0,69
FixMatch		22,09 ± 0,65	77,91 ± 0,65
FlexMatch		21,82 ± 1,17	78,18 ± 1,17
Tamamı Etiketli	Mevcut çalışma	13,00	87,00
Kısmi Etiketli (%4)		44,00	56,00
KKÖ *		38,00	62,00
KKÖ * + Ek Yöntem		36,00	64,00

* KKÖ'nün, ÇKYSA sınıflandırıcı ile birlikte kullanıldığı deneysel çalışmayı gösterir

KAYNAKLAR (REFERENCES)

- [1] J. E. van Engelen ve H. H. Hoos, “A survey on semi-supervised learning”, *Mach Learn*, c. 109, sy 2, ss. 373-440, Şub. 2020, doi: 10.1007/s10994-019-05855-6.
- [2] M.-R. Amini, V. Feofanov, L. Pauletto, E. Devijver, ve Y. Maximov, “Self-Training: A Survey”. arXiv, 15 Şubat 2023. <http://arxiv.org/abs/2202.12040>
- [3] X. Zhu, “Semi-Supervised Learning Literature Survey”, *Comput Sci, University of Wisconsin-Madison*, c. 2, Tem. 2008.
- [4] O. Chapelle, B. Schölkopf, ve A. Zien, Ed., *Semi-supervised learning*. içinde Adaptive computation and machine learning series. Cambridge, Mass. [u.a]: MIT Press, 2010.
- [5] Ke Chen ve Shihai Wang, “Semi-Supervised Learning via Regularized Boosting Working on Multiple Semi-Supervised Assumptions”, *IEEE Trans. Pattern Anal. Mach. Intell.*, c. 33, sy 1, ss. 129-143, Oca. 2011, doi: 10.1109/TPAMI.2010.92.
- [6] I. Triguero, S. García, ve F. Herrera, “Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study”, *Knowl Inf Syst*, c. 42, sy 2, ss. 245-284, Şub. 2015, doi: 10.1007/s10115-013-0706-y.
- [7] O. Chapelle ve A. Zien, “Semi-supervised classification by low density separation”, içinde *International workshop on artificial intelligence and statistics*, PMLR, 2005, ss. 57-64.
- [8] K. Bennett ve A. Demiriz, “Semi-supervised support vector machines”, *Advances in Neural Information processing systems*, c. 11, 1998.
- [9] S. Fralick, “Learning to recognize patterns without a teacher”, *IEEE Trans. Inform. Theory*, c. 13, sy 1, ss. 57-64, Oca. 1967, doi: 10.1109/TIT.1967.1053952.
- [10] A. Blum ve T. Mitchell, “Combining labeled and unlabeled data with co-training”, içinde *Proceedings of the eleventh annual conference on Computational learning theory*, Madison Wisconsin USA: ACM, Tem. 1998, ss. 92-100. doi: 10.1145/279943.279962.
- [11] Q. Xie, M.-T. Luong, E. Hovy, ve Q. V. Le, “Self-training with Noisy Student improves ImageNet classification”. arXiv, 19 Haziran 2020. <http://arxiv.org/abs/1911.04252>
- [12] G. Karamanolakis, S. Mukherjee, G. Zheng, ve A. H. Awadallah, “Self-Training with Weak Supervision”. arXiv, 12 Nisan 2021. <http://arxiv.org/abs/2104.05514>
- [13] G. Tzanetakis, “Automatic Musical Genre Classification of Audio Signals.”, Oca. 2001.
- [14] C. Rosenberg, M. Hebert, ve H. Schneiderman, “Semi-supervised self-training of object detection models”, 2005.
- [15] N. Kamal, M. Andrew, ve M. Tom, “Semi-Supervised Text Classification Using EM”, içinde *Semi-Supervised Learning*, O. Chapelle, B. Scholkopf, ve A. Zien, Ed., The MIT Press, 2006, ss. 32-55. doi: 10.7551/mitpress/9780262033589.003.0003.
- [16] G. Tur, D. Hakkani-Tür, ve R. E. Schapire, “Combining active and semi-supervised learning for spoken language understanding”, *Speech Communication*, c. 45, sy 2, ss. 171-186, Şub. 2005, doi: 10.1016/j.specom.2004.08.002.
- [17] D.-H. Lee, “Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks”, *ICML 2013 Workshop: Challenges in Representation Learning (WREPL)*, Tem. 2013.
- [18] Y. Zou, Z. Yu, B. V. K. V. Kumar, ve J. Wang, “Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training”, 2018, doi: 10.48550/ARXIV.1810.07911.
- [19] P. Cascante-Bonilla, F. Tan, Y. Qi, ve V. Ordonez, “Curriculum Labeling: Revisiting Pseudo-Labeling for Semi-Supervised Learning”. arXiv, 10 Aralık 2020. <http://arxiv.org/abs/2001.06001>
- [20] K. Sohn vd., “FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence”. arXiv, 25 Kasım 2020. <http://arxiv.org/abs/2001.07685>
- [21] P. Yilmaz, Ş. Akçakaya, Ş. D. Özkaya, ve A. ÇetiN, “Machine Learning Based Music Genre Classification and Recommendation System”, *ECJSE*, Ara. 2022, doi: 10.31202/ecjse.1209025.
- [22] S. Sigtia ve S. Dixon, “Improved music feature learning with deep neural networks”, içinde *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy: IEEE, May. 2014, ss. 6959-6963. doi: 10.1109/ICASSP.2014.6854949.
- [23] V. Kiranoglu, G. Tüysüzoğlu, ve E. Öztürk Kiyak, “Prediction of Crime Occurrence in case of Scarcity of Labeled Data”, *Deu Muhendislik Fakultesi Fen ve Muhendislik*, c. 23, sy 68, ss. 677-687, May. 2021, doi: 10.21205/deufmd.2021236828.
- [24] I. Triguero, J. A. Sáez, J. Luengo, S. García, ve F. Herrera, “On the characterization of noise filters for self-training semi-supervised

- in nearest neighbor classification”, *Neurocomputing*, c. 132, ss. 30-41, May. 2014, doi: 10.1016/j.neucom.2013.05.055.
- [25] Y. Wang vd., “USB: A Unified Semi-supervised Learning Benchmark for Classification”. arXiv, 13 Ekim 2022. <http://arxiv.org/abs/2208.07204>
- [26] B. Zhang vd., “FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling”. arXiv, 28 Ocak 2022. <http://arxiv.org/abs/2110.08263>
- [27] B. Zoph vd., “Rethinking Pre-training and Self-training”. arXiv, 15 Kasım 2020. <http://arxiv.org/abs/2006.06882>
- [28] M. Li ve Z.-H. Zhou, “SETRED: Self-training with Editing”, içinde *Advances in Knowledge Discovery and Data Mining*, c. 3518, T. B. Ho, D. Cheung, ve H. Liu, Ed., içinde Lecture Notes in Computer Science, vol. 3518. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ss. 611-621. doi: 10.1007/11430919_71.
- [29] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, ve J. Wang, “Confidence Regularized Self-Training”. arXiv, 15 Temmuz 2020. <http://arxiv.org/abs/1908.09822>
- [30] A. Krizhevsky, G. Hinton, ve others, “Learning multiple layers of features from tiny images”, 2009.
- [31] H. Schmutz, O. Humbert, ve P.-A. Mattei, “Don’t fear the unlabelled: safe semi-supervised learning via simple debiasing”. arXiv, 03 Mart 2023. <http://arxiv.org/abs/2203.07512>
- [32] A. Olteanu, “GTZAN Dataset-Music Genre Classification. Kaggle. com (2019)”. <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>
- [33] T. Hastie, J. Friedman, ve R. Tibshirani, *The Elements of Statistical Learning*. içinde Springer Series in Statistics. New York, NY: Springer New York, 2001. doi: 10.1007/978-0-387-21606-5.
- [34] T. Cover ve P. Hart, “Nearest neighbor pattern classification”, *IEEE Trans. Inform. Theory*, c. 13, sy 1, ss. 21-27, Oca. 1967, doi: 10.1109/TIT.1967.1053964.
- [35] B. Charbuty ve A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning”, *JASTT*, c. 2, sy 01, ss. 20-28, Mar. 2021, doi: 10.38094/jastt20165.
- [36] S. J. Rigatti, “Random Forest”, *Journal of Insurance Medicine*, c. 47, sy 1, ss. 31-39, Oca. 2017, doi: 10.17849/in-sm-47-01-31-39.1.
- [37] M. Belgiu ve L. Drăguț, “Random forest in remote sensing: A review of applications and future directions”, *ISPRS Journal of Photogrammetry and Remote Sensing*, c. 114, ss. 24-31, Nis. 2016, doi: 10.1016/j.isprsjprs.2016.01.011.
- [38] T. Chen ve C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, Mar. 2016.
- [39] D. Mustafa Abdullah ve A. Mohsin Abdulazeez, “Machine Learning Applications based on SVM Classification A Review”, *QAJ*, c. 1, sy 2, ss. 81-90, Nis. 2021, doi: 10.48161/qaj.v1n2a50.
- [40] M. Maalouf, “Logistic regression in data analysis: an overview”, *IJDATS*, c. 3, sy 3, s. 281, 2011, doi: 10.1504/IJDATS.2011.041335.
- [41] J. Zou, Y. Han, ve S.-S. So, “Overview of Artificial Neural Networks”, içinde *Artificial Neural Networks*, c. 458, D. J. Livingstone, Ed., içinde *Methods in Molecular Biology™*, vol. 458. , Totowa, NJ: Humana Press, 2008, ss. 14-22. doi: 10.1007/978-1-60327-101-1_2.
- [42] Nart Sooksil ve Vacharapoom Benjaoran, “Non-linear modelling of construction workers’ behaviors for accident prediction”, *Songklanakarin Journal of Science and Technology (SJST)*, c. 43, s. 596602, 2021, doi: 10.14456/SJST-PSU.2021.80.