Fırat Üniversitesi Fen Bil. Dergisi
34(1), 19-29, 2022

Araştırma Makalesi

Türkçe Ses Kayıt Verilerinin CountVectorizer ve TF-IDFVectorizer Yöntemleri ile BERT Modelleri Olarak Google Colab Platformunda ve RapidMiner'da Makine Öğrenmesi Algoritmalarıyla Analizi

Abdülkadir TEPECİK[1*], Engin DEMİR[2]
[1,2] Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Yalova Üniversite, Yalova, Türkiye
[*1] atepecik@yalova.edu.tr, [2] engindmr7@gmail.com

(Geliş/Received: 11/09/2021; Kabul/Accepted: 01/01/2022)

Öz: Duygu, insanın ruh halinde içsel ve çevresindeki etkilerle etkileşiminden doğan fiziksel değişimlerdir. Bireyler duygularını, beden dilinin yanı sıra sesli iletişim vasıtalarıyla da diğer bireylere aktarabilirler. Özellikle beden dilinin yetersiz olduğu durum ve zamanlarda bireyler için sesli iletişim önem kazanmaktadır. Çalışmamızda da Türkçe ses kayıtlarını içeren veri seti üzerinde Python programlama dili aracılığıyla öncelikle verilerin duygu etiketlerinin tespiti yapılmış olup, sonrasında literatür çalışmalarında en çok kullanılan beş makine öğrenmesi algoritmasıyla analizler gerçekleştirilmiştir. Belirlenen metriklerle yapılan analizler hem RapidMiner hem de Python programlama dili aracılığıyla gerçekleştirilmiştir. Çalışmada, Python programlama dili aracılığıyla yapılan analizlerde hem CountVectorizer hem de TF-IDFVectorizer yöntemleri, RapidMiner ile yapılan analizlerde TF-IDFVectorizer yöntemi kullanılmıştır. Sonuç kısmında ise Python programlama dilinde en iyi doğruluk oranını %70 oranla Naive Bayes makine öğrenmesi algoritması CountVectorizer yöntemiyle elde etmiştir. RapidMiner'da ise en iyi doğruluk oranını %69,60 oranla Destek Vektör Makinesi makine öğrenmesi algoritması elde etmiştir. Çalışmamızla beraber ortaya yeni bir Türkçe duygu veri seti çıkmıştır. Ayrıca çalışmamız Türkçe ses kayıtlarından elde edilen verilerin BERT modeli ile duygu tespiti yapılan özgün bir çalışmadır.

Anahtar kelimeler: Türkçe Ses Kayıtları, Türkçe Duygu Analizi, RapidMiner, Python, Makine-Öğrenmesi.

Analysis of Turkish Voice Recordings Data with CountVectorizer and TF-IDF Vectorization Methods as BERT Models on Google Colab Platform and RapidMiner with Machine Learning Algorithms

Abstract: Emotions are physical changes in a person's mood resulting from his interaction with internal and environmental influences. Individuals can convey their feelings to other individuals by means of voice communication as well as body language. Especially in situations and times when body language is insufficient, voice communication becomes important for individuals. In our study, the emotion labels of the data were firstly determined by the Python programming language on the data set containing the Turkish voice recordings, and then the analyses were carried out with the five most used machine learning algorithms in the literature studies. The analyses made with the determined metrics were carried out using both RapidMiner and Python programming language. In the study, both CountVectorizer and TF-IDF vectorization methods were used in analyses performed through the Python programming language, and TF-IDF vectorization method was used in analyses performed with RapidMiner. As a result, the best accuracy rate in the Pyhton programming language was achieved by the Naive Bayes machine learning algorithm CountVectorizer method with 70%. At RapidMiner, the Support Vector Machine machine learning algorithm achieved the best accuracy rate of 69.60%.with our study, a new Turkish emotion dataset has emerged. In addition, our study is an original study in which emotion detection is made with the BERT model of the data obtained from Turkish audio recordings.

Key words: Turkish Auido Recordings, Turkish Sentiment Analysis, RapidMiner, Pyhton, Machine-Learning.

1. Giriş

İnsanların ruhsal etkenlerle ve çevrelerindeki olaylara göre verdikleri tepkilerin tanımına his veya duygu denir [1]. Duyguyu ifade etmenin birçok yolu vardır. Bunlardan biri beden dilinin yetersiz kaldığı durumlardaki ifade yöntemi olan sesteki duygu ifadesidir.

İnsanların olumlu ve olumsuz duygularını rahatlıkla ifade edebilmeyi imkân buldukları andan itibaren duygu analizine ilgi gün geçtikçe çoğalmış ve bu konudaki çalışma sayısı artmıştır. Duygu tespiti, duygu analizi gibi terimler, makine öğrenmesiyle beraber daha çok ön plana çıkmış ve veri bilimi ile ilgili araştırma alanlarının odak noktası haline gelmiştir. Duygu tespiti sonrası makine öğrenmesi algoritmaları ile yapılan analizler, yapılacak olan birçok çalışma için de yeni fikirlerin ortaya çıkmasına katkı sağlamıştır.

[*] Sorumlu yazar: atepecik@yalova.edu.tr. Yazarların ORCID Numarası: [1] 0000-0002-3842-7873, [2] 0000-0002-4546-3581

Makine öğrenmesi literatürde, *insan zekâsını ve algısını taklit ederken, kişinin yorumlayıp elle gireceği kurallara* ihtiyaç *duymayan algoritmalar bütünü* olarak ifade edilmektedir [2].

Yapılan çalışmalarda; S. Tuzcu, çevrimiçi bir kitap satış sitesinin kullanıcı yorumları üzerinde duygu analizi için öncelikle Pyhton programlama dili kullanarak Çok Katmanlı Algılayıcı (Multi-Layer Perceptron MLP) algoritması kullanmayı amaçlamıştır. Çalışmasında, Naive Bayes (NB), Destek Vektör Makinesi (DVM) ve Lojistik Regresyon (LR) algoritmaları kullanılmış ve sonuçları karşılaştırılmıştır. Sonuçlarda; DVM, pozitif yorumları sınıflandırmada tüm algoritmalar içinde en iyi neticeyi vermiş fakat negatif yorumları sınıflandırmada diğer algoritmalardan oldukça geride kalmıştır. NB ise bu üç algoritmadan farklı olarak negatif yorumları pozitif yorumlara göre daha iyi sınıflandırmış fakat genel sınıflandırma başarısı bu algoritmaların altında kalmıştır [3].

J. Gondohanindijo ve arkadaşları, duygusal konuşmayı tanıma problemini çözmeyi, ses işleme yoluyla sınıflandırmayı, duyguların sınıflandırılmasına dayanan ses kalıplarını tanımayı ve kullanılan bazı sınıflandırıcıların doğruluk düzeyini karşılaştırmayı amaçlamışlardır[4]. Çalışmada 639 veriden oluşan veri seti üzerinde DVM, Rastgele Orman (RO), NB ve Sinir Ağı sınıflandırıcıları kullanılmıştır. DVM sınıflandırıcısının en iyi doğruluk ve hassasiyet seviyesine sahip olduğu ve F1-skorunda diğer üç sınıflandırıcıya göre daha iyi olduğu belirtilmiştir.

O. M. Nezami ve arkadaşları, Farsça için ShEMO adı verilen duygusal konuşma veri tabanı sunmuşlardır. Veri tabanında sınıflandırıcılar karşılaştırılarak, çıkan sonuçlar diğer dillerle kıyaslanmıştır[5]. Veri tabanı, 3 saat 25 dakikalık konuşma verilerine eşdeğer 3000 yarı doğal ifadeyi ve beş duyguyu içermektedir. Çalışmada, sınıflandırma algoritmalarından DVM, K-En Yakın Komşu (KEYK) ve Karar Ağacı (KA) kullanılmıştır. DVM modelinin en iyi sonuçları verdiği, gelecekteki çalışmalar için Gizli Markov Modeli ve Derin Sinir Ağları gibi sınıflandırma yöntemlerini kullanarak daha etkili bir tespit elde etmeyi amaçladıklarını belirtmişlerdir.

R. Matin ve D. Valles, otizm spektrum bozukluğu olan çocuklara, sosyal etkileşimdeki duyguları tanımlamak için bir konuşma duygu tanıma modeli geliştirmeyi amaçlamışlardır[6]. DVM modeline dayalı önerilen model Pyhton programlama dili ile geliştirilmiştir. Çalışmada RAVDESS veri setiyle beraber DVM, Karar Ağacı ve LR sınıflandırma algoritmaları kullanılmıştır. Yapılan deneylerin sonucunda DVM modelinin %77 oranda doğrulukta ve önerilen modelin arka plan gürültüsü ile ses için iyi sınıflandırma doğruluğu sağladığı belirtilmiştir.

S. Dani ve arkadaşları, seslerdeki duyguyu tespit etmeyi amaçlamışlardır[7]. KEYK ve Karar Ağacı makine öğrenmesi tekniklerini, Toronto Duygusal Konuşma Seti (TESS) adlı veri seti üzerinde 7 duygu için uygulamışlardır. KEYK tekniğinin %98, Karar Ağaçlarının %92 ve Ekstra-Ağaç sınıflandırıcısının %99 doğruluk oranı sağladığını belirtmişlerdir.

M. G. Pinto ve arkadaşları, sesteki duygu analizinin tespiti için Konvolusyonel Sinir Ağlarına dayanan bir sınıflandırma modeli sunmuşlardır[8]. Çalışmada 8 duygunun tespiti için RAVDESS veri seti kullanılmıştır. F1-skoruna göre değerlendirme yapılmış, en yüksek skorun kızgın duygusuna, en düşüğün ise üzgün duygusuna ait olduğunu belirtmişlerdir.

Bu makalede Türkçe bir duygu veri tabanı oluşturmak, kullanılan metot ve materyallerle yapılacak olan çalışmalara öncü olmak, duygu tespitinde kullanılan BERT modelinin Türkçe yapısını geliştirmek ve makine öğrenmesi algoritmalarının Türkçe dil yapısı üzerindeki başarı oranlarını ölçmek amaçlanmıştır.

2. Veri seti ve Yöntemler

Veri seti, duygu tespiti için kullanılan BERT modeline ve analiz için kullanılan makine öğrenmesi algoritmaları aşağıdaki alt bölümlerde verilmiştir.

2.1. Veri seti

Veri seti, Mozilla tarafından başlatılan ses ve konuşma tanıma yazılımları için ücretsiz bir veri tabanı oluşturulmak üzere geliştirilen Common Voice platformundan alınmıştır. İnsanların nasıl konuştuklarını makinelere öğretmek amacıyla oluşturulan bu platformda çeşitli diller için ses kayıtları ve bu ses kayıtlarına ait metin dokümanları bulunmaktadır. Türkçe ses kayıtlarını içeren 592 MB boyutundaki dosya yaklaşık 20.760 ses verisini içermektedir. Ses kayıtlarının içerikleri ise; haber videolarından elde edilen sesler ve platform üzerinden yapılan kayıtlar oluşturmaktadır. Bu veri setine ait bilgiler ise Tablo 1'de verilmiştir.

Tablo 1. Veri seti yaş aralıkları ve oranları

Yaş Aralığı	Oranı
<19	%4
19-29	%47
30-39	%23
40-49	%3
50-59	%1

Bu veri seti için %71 erkek , %6 kadın şeklinde bir cinsiyet oranı bilgisi de paylaşılmıştır. Birçoğu tekrar eden kayıtlarından oluşan bu ses kayıtlarının tüm metin dokümanları birleştirilmiştir. Elde edilen metin dokümanları üzerinde eleme işlemi gerçekleştirilmiştir. Bir anlam ifade etmeyen, tekrar eden ve sadece sayı belirten veriler ayrıştırıldıktan sonra 5001 adet metin verisi elde edilmiştir. Bu veri seti, üzerinde anlamsız verilerin ayrıştırılmasından sonra çalışmanın içeriğini oluşturan duygu tespiti ve analiz işlemlerine hazır hale getirilmiştir. Çalışmanın deney kısımlarında veri setinin bir kısmı, öğrenme-test için kullanılmıştır. Denemeler sonucu test veri seti oranı %15 olarak belirlenmiştir. Veri seti üzerinde, verilerin duygu etiketleri BERT modeli aracılığıyla negatif ve pozitif olarak belirlenmiştir. Literatür taramasında en çok kullanılan beş makine öğrenmesi algoritması belirlenmiştir. Bu algoritmalar; Naive Bayes, Destek Vektör Makinesi, Rastgele Orman, Karar Ağacı ve K-En Yakın Komşu'dur. Python programlama dili aracılığıyla gerçekleştirilen çalışmada yapılan analizlerin daha geniş çerçevede değerlendirilmesi için hem CountVectorizer hem de TF-IDFVectorizer yöntemleri kullanılmıştır. Her iki vektörizasyon yöntemi, çalışmadaki makine öğrenmesi algoritmalarının hepsinde kullanılmıştır. RapidMiner platformuyla yapılan analizler ise sadece TF-IDFVectorizer yöntemi ile değerlendirilmiştir.

Sonuçlar ise doğruluk (accuracy), duyarlılık değeri (precision), anma değeri (recall) ve F1-skoru (F1-score) gibi metriklerle değerlendirilmiştir. Çalışma hem RapidMiner platformunda hem de Python programlama dili ile Google'ın Colab ortamında gerçekleştirilmiştir.

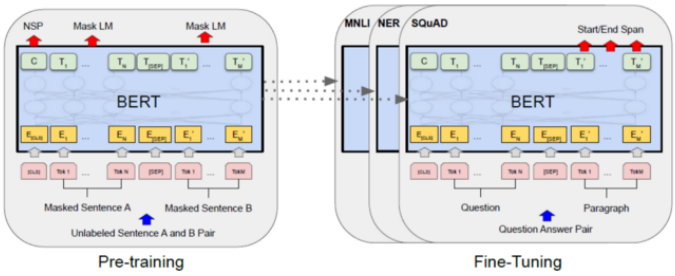
2.2. Metin Verilerini Kodlama Yöntemleri

Veri setinde bulunan cümleler üzerinde işlem yapabilmek için belirteçleme (tokening) adı verilen yöntemle cümledeki kelimelerin ayrıştırılması gerekir. Daha sonra makine öğrenim algoritmasına giriş olarak kullanılmak üzere kelimelerin, vektörleştirme (özellik ayıklama feature-extraction) işlemi ile tamsayılar veya kayan nokta değerleri olarak kodlanması gerekir. Klasik olarak 3 metot kullanılmaktadır. Bunlar; metni kelime sayısı vektörlerine dönüştüren CountVectorizer, kelime frekansı vektörlerine dönüştüren TF-IDFVectorizer ve benzersiz tamsayılara (hashing) dönüştüren HashingVectorizer'dir.

Bu çalışmada ilk iki sayısallaştırma metodu kullanılmıştır.

2.3. BERT

BERT (Transformatörlerden Çift Yönlü Kodlayıcı Temsilleri), çok çeşitli doğal dil işleme görevleri hakkında en gelişmiş sonuçları elde eden yeni bir eğitim öncesi dil temsili yöntemidir. 2018 yılında Google'da Jacob Devlin ve arkadaşları tarafından geliştirilen bu model yapısı cümleyi hem sağdan sola hem de soldan sağa değerlendirmektedir. Böylelikle kelimelerin birbirleriyle olan ilişkilerini daha iyi ortaya koymaktadır. Modelin yapısında, BookCorpus ve Wikipedia veri setleri bulunmaktadır[9]. BERT model yapısı Şekil 1'de verilmiştir.



Şekil 1. BERT model yapısı [10]

Model, Masked Language Modeling (MLM) ve Next Sentence Prediction (NSP) yöntemleri ile eğitilmektedir. MLM tekniğinde, maskelenen kelime, açık şekilde beslenen kelimelerden yararlanılarak tahmin edilmeye çalışılır. Daha çok kelimeler arasında ilişkiler üzerinde durulmaktadır. Bir diğer yöntem NSP'de ise cümleler arasındaki ilişki üzerinde durulmaktadır. NSP ile model eğitim aşamasındayken iki cümle yapısı arasındaki ilişkiye bakılarak ikinci cümlenin ilkinin devamı olup olmadığına bakılır ve tahmin yapılmaya çalışılır. Eğitim aşamasında gerçekleştirilen optimizasyon ile bu iki yöntem kullanılırken ortaya çıkan kaybın minimuma indirilmesi amaçlanmıştır [9]. BERT model yapısı verilen metin dokümanı hem sağdan hem soldan incelemesi, yapısında bulunan yöntemlerle eğitim aşamasında daha iyi bir öğrenme sağlamaktadır.

2.4. Naive Bayes

Naive Bayes algoritması, Bayes teoremine dayalı bir öğrenme algoritmasıdır. Tembel yapıya sahip olmasına rağmen düzensiz yapıdaki veri setlerinde de çalışabilmektedir. Bayes Teoreminin matematiksel ifadesi (1) gibidir

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

P(A|B); B olayı gerçekleştiği durumda A olayının meydana gelme olasılığı, P(B|A) ; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığı ve P(A) ve P(B) ; A ve B olaylarının olasılıklarıdır.

NB sınıflandırma algoritması, tüm koşullu olasılıkların çarpımıdır [12]. Bu sınıflandırma algoritmasıyla verilen eğitim ve test verilerinin doğru bir şekilde sınıflandırılması yapılabilmektedir. NB sınıflandırma algoritmasına ait formül (2)'de verilmiştir.

$$P(Class_j \mid x) = \frac{P(x \mid Class_j)P(Class_j)}{P(x)} \tag{2}$$

Burada verilmiş bir x tahminci olasılığı, j sınıfına ait olasılık ve eğitim kümesinden bulunan bu sınıf tahminci olasılığı ile x tahmincisinin bu sınıfa ait olma olasılığı hesaplanabilir. Bu algoritma spam filtreleme, duyarlılık analizi, çok sınıflı tahmin, öneri sistemleri ve benzeri alanlarda kullanılmaktadır [11].

2.5. Rastgele Orman

Rastgele Orman, birden fazla karar ağacının avantajlarını kullanarak daha uyumlu ve daha iyi sonuçlar üreten modeller ortaya koyarak sınıflandırma işlemini en iyi şekilde yapmaya çalışan bir sınıflandırma algoritmasıdır. RO çoğu zaman büyük bir sonuç üreten, esnek, kullanımı kolay bir makine öğrenmesi algoritmasıdır. RO algoritmasının en önemli avantajı ise hem sınıflandırma hem de regresyon işlemlerinde kullanılabilmesidir[13]. RO algoritmasına ait matematiksel ifade (3)'te verilmiştir. (3) denkleminde $fi\ sub(i):\ i$ özelliğinin önemi ve $ni\ sub(j):\ j$ düğümünün önemi olmak üzere;

$$fi_i = \frac{\sum_{j:i\ özelligine\ ayrilan\ j\ dugumu} ni_j}{\sum_{k \in bütün\ dügümler} ni_k} \tag{3}$$

Bunlar bütün özellik önem değerlerinin toplamına bölünerek 0 ve 1 arasında bir değere normalize edilirse (4) ifadesi yazılabilir yani;

$$norm\ fi_i = \frac{fi_i}{\sum_{j \in bütün\ özellikler} fi_j} \tag{4}$$

RO düzeyindeki nihai özelliğin önemi, tüm ağaçların ortalamasıdır. Her ağaçtaki özelliğin önem değerinin toplamı hesaplanır ve toplam ağaç sayısına bölünürse (5) denklemi elde edilir:

$$RFfi_i = \frac{\sum_{j \in bütün\ agaclar} normfi_{ij}}{T} \tag{5}$$

Burada, $RFfi_i$ RO modelindeki tüm ağaçlardan hesaplanan özelliğinin önemi, $norm\ fi_{ij}$ j ağacındaki i için normalleştirilmiş özellik önemi, T ise toplam ağaç sayısıdır. Bu algoritma aynı zamanda sınıflandırıcı kategorik veriler için de değerlendirilebilmektedir. RO algoritması ayrıca over-fitting (aşırı uyum) problemlerini de gidermektedir.

2.6. Destek Vektör Makinesi

Destek Vektör Makinesi, 1995 yılında Vapnik tarafından istatiksel öğrenme teorisi ve VC-boyut teorisinden türetilmiştir. DVM, ikili sınıflandırma problemini çözmek için geliştirilmiştir. DVM, karmaşık verilerin yüksek doğrulukla işleme özelliği olarak da tanımlanmaktadır. DVM'deki esas amaç eğitilmiş veri örneklerini önceden tanımlanmış sayıda sınıfa ayıran hiper düzlem bulmaktır [14-15]. DVM'nin matematiksel ifadeleri (6) ve (7) de verilmiştir.

$$\begin{aligned} &\vec{x}_i * \vec{w} + b = 1; \text{ pozitif siniflar icin} \\ &\vec{x}_i * \vec{w} + b = -1; \text{ negatif siniflar icin} \end{aligned} \tag{6}$$

Sabit marjlı DVM, hiper düzlemi geçen destek vektörleri konusunda çok kesindir. Hiper düzlemin marjını maksimize etmek için, sabit marjlı DVM optimizasyon problemine dönüşür yani;

$$\begin{aligned} &\min_{\vec{w},b} \frac{1}{2}\|\vec{w}\|^2 \\ &y_i(\vec{x}_i.\vec{w} + b) \geq 1; \\ &i = 1,...,n \end{aligned} \tag{7}$$

Bu sınıflandırma algoritmasının avantajları olarak şunlar söylenebilir; yüksek boyutlu uzayda diğer algoritmalara göre daha etkinlerdir, amaç fonksiyonu için farklı çekirdek fonksiyon yapıları kullanmaktadırlar ve amaç noktalarında kullandıkları eğitim fonksiyonları sayesinde belleğin daha verimli kullanılmasına olanak sağlamaktadırlar [16].

Algoritma kendi içinde de doğrusal ve doğrusal olmayan olarak ikiye ayrılmaktadır. Doğrusal olmayan yöntemde çekirdek fonksiyonları yöntemi kullanılmaktadır. DVM algoritması, metin ve görüntü sınıflandırmada, biyolojik bilim alanlarında ve elle yazılmış karakterlerin tanınması gibi alanlarda kullanılmaktadır.

2.7. Karar Ağacı

Karar Ağacı, özellik, hedef ve sınıflamaya göre karar düğümlerinden ve yaprak düğümlerinden oluşan ağaç şeklinde bir model yapısı oluşturan gözetimli sınıflandırma algoritmasıdır [17]. Yapısında bulunan en üst düğüme root (kök) ve diğer düğüm yapılarına da leaf (yaprak) adı verilmektedir. Karar Ağacı algoritmaları veri setini küçük parçalara bölerek geliştirilmektedir. Böylelikle algoritma yapısında büyük kayıpların önüne geçilerek daha küçük kayıpların olması sağlanmıştır.

Algoritma yapısı, ağaç yapısına benzetildiği için kullanılan matematiksel ifadelerde de bu durum göz önünde bulundurulmuştur. Belirli bir T=t ağacı ve D=d eğitim veri seti için, t'nin d üzerinde ne kadar iyi çalıştığına dair olasılıksal tahmin yani P(T=t|D=d) bulunabilir. İdeal ağaç ise maksimum P(T=t|D=d) değerine sahip olmalıdır.

$$P(Y = y1 \mid X = x, D = d) = \sum_{BütünAgaclar} P(Y = y1 \mid T = t, X = x, D = d) P(T = t \mid D = d) \tag{8}$$

Buna göre, bir ağaç yapısı eğitim verilerini ne kadar iyi ayırırsa P(T=t|D=d) ifadesi de maksimum değeri alır.

Karar ağaçlarının avantajları ise şöyledir; hem kategorik hem de sayısal verileri işleyebilirler, birden fazla çıktısı olan problemler için de çözüm üretebilirler, yorumlanması ve anlaşılması kolaydır ve kullanılan ağaç yapıları görsel halinde sunulabilir [18].

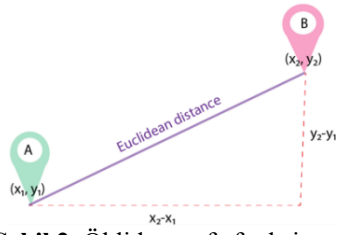
2.8. K-En Yakın Komşu

K-En Yakın Komşu algoritması, eklenecek olan yeni verinin mevcut veri kümelerine uzaklığını hesaplayarak, K sayıda yakın komşuluğa bakarak sınıflandırma işlemini gerçekleştirmektedir. Uzaklık hesaplamaları için, Öklid (Euclidean), Minkowski ve Manhattan uzaklık hesaplamalarını kullanmaktadır. Algoritma yapısı eski, gürültülü eğitim verilerine karşı dirençli olduğundan günümüzde halen kullanılmakta olan popüler bir sınıflandırma algoritmasıdır. Algoritmanın işleyişi ise şu şekildedir; ilk olarak k parametresi belirlenir, uzaklık hesaplamaları gerçekleştirilir, komşular bulunur ve veri etiketlemesiyle süreç tamamlanır.

Algoritmanın hesaplamalarında kullanılan Öklid mesafe fonksiyonuna göre (x_1, y_1) ve (x_2, y_2) noktaları arasındaki mesafe (9)'daki formüle göre hesaplanır.

$$dist((x_2, y_2),(x_1, y_1)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{9}$$

Şekil 2'de bu formülün iki boyutta grafiği verilmiştir.



Şekil 2. Öklid mesafe fonksiyonu

KEYK algoritmasında, bir K değeri için, algoritma veri noktasının K'ya en yakın komşularını en fazla veriye sahip noktalarına atar. Matematiksel olarak denklem (10)'daki gibi verilir.

$$P(y = j \mid X = x) = \frac{1}{K} \sum_{i \in A} I(y^{(i)} = j) \tag{10}$$

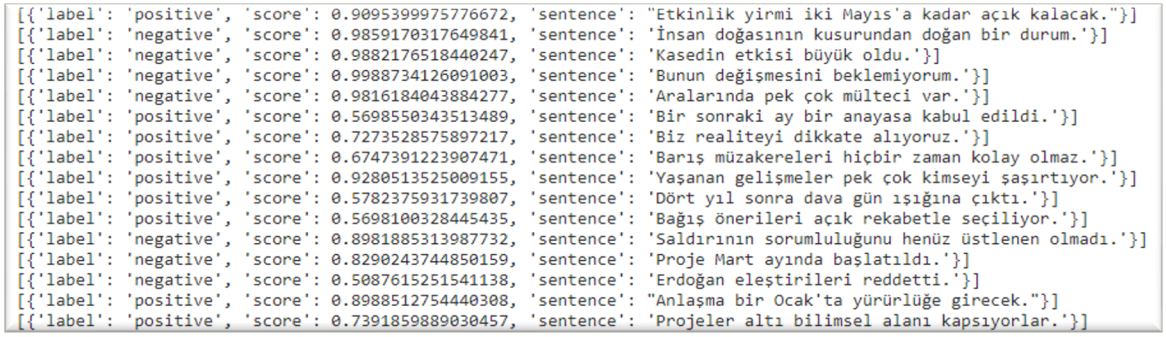
KEYK algoritması, hem regresyon hem de birçok alanda sınıflandırma işlemlerinde kullanılmaktadır [19].

3. Uygulamalar ve Sonuçları

3.1. Google Colab'da BERT Modeli ile Uygulama

Türkçe duygu tespitinde BERT modelindeki gibi cümleler pozitif ve negatif olarak değerlendirilmektedir. İlk olarak gerekli kütüphanelerin ve veri setinin yüklenmesi gerçekleştirilmiştir. Daha sonra cümle yapıları belirteçlerine ayrıldıktan sonra modele aktarılır ve burada cümlelerin pozitiflik veya negatiflik skorları ve etiketleri belirlenir. Kod kısmında pipeline nesnesi aracılığıyla veriler, peş peşe seri biçimde tanımlanmış birtakım işlemlerden geçmektedir. Bunlar sırasıyla verilerin tokenlerine ayrılması ve sonrasında yüklenen BERT modeline aktarılmasıdır.

Veri setinde bulunan cümlelerin karşısında duygu etiketleri (pozitif - negatif) ve skorları yazdırılmıştır. Gerçekleştirilen duygu tespiti sonrasında 5001 cümleden 2956 negatif ve 2045 pozitif duygu etiketine sahip cümle elde edilmiştir. Veri seti içinde oransal olarak %59 negatif ve %41 pozitif cümle bulunmaktadır. Çıktı sonuçlarının bir kısmı Şekil 3'te verilmiştir.



Şekil 3. BERT modeli ile yapılan duygu tespiti sonuçları

Duygu etiketleri belirlenen verilerden oluşan veri seti üzerinde makine öğrenmesi algoritmalarıyla belirlenen metrikler üzerinden analizler gerçekleştirilmiştir. Bu analizler, Python programlama dili aracılığıyla hem CountVectorizer hem de TF-IDFVectorizer yöntemiyle yapılmıştır.

Her iki yöntemde, çalışmada kullanılan tüm makine öğrenmesi algoritmalarıyla uygulanmıştır. Veri seti, %85 eğitim verisi ve %15 test verisi olarak ayrılmıştır. Analizler test veri seti üzerinde gerçekleştirilmiştir. Test veri setinde 431 negatif, 320 pozitif olmak üzere 751 veri bulunmaktadır.

Tablo 2'deki sonuçlarda duyarlılık değeri, anma değeri, F1-skoru ve doğruluk metriklerine ait sütunlar ikiye ayrılmıştır. Sol taraftaki veriler CountVectorizer yöntemine, sağ tarafta bulunan veriler ise TF-IDFVectorizer yöntemiyle elde edilen sonuçlara aittir. Mavi renk ile belirtilen hücrelerdeki değerler negatif veriler arasında en yüksek değeri, turuncu renkteki hücreler ise pozitif veriler arasındaki en yüksek değeri belirtmektedir.

Tablo 2. CountVectorizer (sol) ve TF-IDFVectorizer (sağ) yöntemleriyle elde edilen sonuçlar

Algoritmalar	**Veri Duygu Durumu**	**Duyarlılık Değeri**		**Anma Değeri**		**F1-Skoru**		**Doğruluk**	
Naive Bayes	Pozitif	0.68	0.71	0.55	0.44	**0.61**	0.54	**%70**	%68
	Negatif	**0.71**	0.68	0.81	0.87	**0.75**	0.76		
RO	Pozitif	**0.69**	0.64	0.42	0.49	0.52	0.56	%67	%67
	Negatif	0.66	0.68	0.86	0.80	**0.75**	0.73		
DVM	Pozitif	0.63	0.67	**0.56**	**0.56**	0.59	**0.61**	%68	**%69**
	Negatif	0.70	**0.71**	0.76	0.79	0.73	**0.75**		
KA	Pozitif	0.59	0.54	0.42	0.51	0.49	0.52	%63	%61
	Negatif	0.65	0.65	0.78	0.68	0.71	0.67		
KEYK	Pozitif	0.68	**1.00**	0.17	0.07	0.27	0.13	%61	%60
	Negatif	0.60	0.59	**0.94**	**1.00**	0.74	0.74		

Tabloya göre;

Doğruluk oranında en iyi sonucu NB algoritması CountVectorizer ile %70 olarak elde etmiştir. En düşük doğruluk oranını ise %60 ile KEYK algoritması, TF-IDFVectorizer ile elde etmiştir.

Duyarlılık değeri için ilk değerlendirme negatif veriler üzerinde ve CountVectorizer ile elde edilen sonuçlar için yapılmıştır. %71 ile en iyi duyarlılık değeri oranını NB elde etmiştir. TF-IDFVectorizer ile en iyi duyarlılık oranını %71 ile DVM elde etmiştir. Pozitif veriler için ise %69 ile en iyi duyarlılık oranını RO elde etmiştir. TF-IDFVectorizer ile en iyi duyarlılık oranını %100 oranla KEYK elde etmiştir.

Anma değeri için ilk değerlendirme negatif veriler üzerinde ve CountVectorizer ile elde edilen sonuçlar için yapılmıştır. %94 ile en iyi anma değeri oranını KEYK elde etmiştir. TF-IDFVectorizer ile en iyi anma değeri oranını %100 oranla (1.00) KEYK elde etmiştir. Pozitif veriler için ise %56 ile en iyi anma değeri oranını DVM elde etmiştir. TF-IDFVectorizer ile en iyi anma değeri oranını %56 oranla DVM elde etmiştir.

F1-skoru için ilk değerlendirme negatif veriler üzerinde ve CountVectorizer ile elde edilen sonuçlar için yapılmıştır. %75 ile hem NB hem de RO elde etmiştir. TF-IDFVectorizer ile en iyi F1-skorunu %71 oranla DVM elde etmiştir. Pozitif veriler için ise %61 ile en iyi F1-skoru oranını NB elde etmiştir. TF-IDFVectorizer ile en iyi F1-skorunu %61 oranla DVM elde etmiştir.

Makine öğrenmesi algoritmaları ile yapılan analizden karmaşıklık matrisleri de elde edilmiştir.

Karmaşıklık matrisi, Python programlama dilinde yapılan analizle her iki vektörizasyon işlemi sonucunda da elde edilmiştir. Karmaşıklık matrisinde negatif ve pozitif duygu etiketlerinde gerçek değerler ve tahmin değerleri gösterilmiştir. Sonuçlar Tablo 3'te verilmiştir.

Tablo 3. CountVectorizer (sol) ve TF-IDFVectorizer (sağ) yöntemlerinin karmaşıklık matrislerine ait sonuçlar

Algoritmalar		**Veri Duygu Durumu**	**Tahmin Edilen Değerler**			
			Negatif		Pozitif	
NB	**Gerçek Değerler**	Negatif	348	374	83	57
		Pozitif	143	180	177	140
RO		Negatif	371	343	60	88
		Pozitif	187	162	133	158
DVM		Negatif	328	341	103	90
		Pozitif	141	141	179	179
KA		Negatif	337	295	94	196
		Pozitif	184	158	136	162
KEYK		Negatif	405	431	26	0
		Pozitif	266	297	54	23

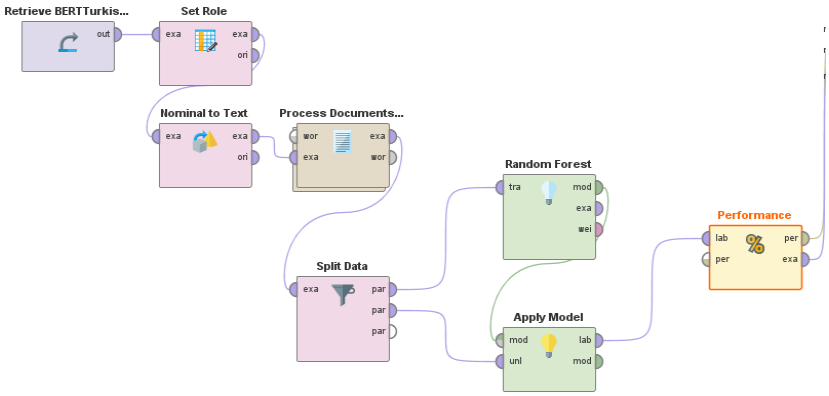
Özet olarak en iyi sonuçlara ulaşan algoritmalar Tablo 4'te verilmiştir.

Tablo 4. CountVectorizer (sol) ve TF-IDFVectorizer (sağ) ile elde edilen sonuçlarda en iyi oranlara ulaşan algoritmalar

Veri Duygu Durumu	**Duyarlılık Değeri**		**Anma Değeri**		**F1-Skoru**		**Doğruluk**	
Negatif	NB	DVM	KEYK	KEYK	NB, RO	DVM	NB	DVM
Pozitif	RO	KEYK	DVM	DVM	NB	DVM		

3.2. RapidMiner'da Uygulama

İkinci uygulama olarak RapidMiner platformuyla, aynı veriler üzerinde analizler gerçekleştirilmiştir. Vektörizasyon yöntemi olarak sadece TF-IDFVectorizer yöntemi kullanılmıştır. Veri seti, %85 eğitim verisi ve %15 test verisi olarak ayrılmıştır. Analizler test veri seti üzerinde gerçekleştirilmiştir. Her makine öğrenmesi algoritması için operatörlerden oluşan model yapısı oluşturulmuştur. Bu model yapısında her seferinde sadece kullanılan makine öğrenmesi algoritması değişmiştir. Örnek olarak RO algoritmasına ait model, Şekil 4'te verilmiştir.



Şekil 4. RapidMiner'da RO algoritması için oluşturulan model yapısı

Oluşturulan model yapıları sonrasında çıkan sonuçlar Tablo 5'te verilmiştir.

Tablo 5. TF-IDFVectorizer yöntemiyle elde edilen sonuçlar

Algoritmalar	Veri Duygu Durumu	Duyarlılık Değeri	Anma Değeri	Doğruluk
NB(Kernel)	Pozitif	%62.35	%51.79	%67.47
	Negatif	%70.10	%78.33	
RO	Pozitif	%0	%0	%59.07
	Negatif	%59.07	%100	
DVM	Pozitif	%90.29	%39.74	%69.60
	Negatif	%68.38	%73.94	
KA	Pozitif	%85.71	%3.91	%60.40
	Negatif	%59.92	%99.55	
KEYK	Pozitif	%64.94	%53.09	%69.07
	Negatif	%71.14	%80.14	

Pozitif ve negatif duygu etiketine sahip olan veri seti üzerinde yapılan işlemler sonrasında en yüksek doğruluk oranını DVM elde etmiştir. Doğruluk oranı %69.60'dır. Analiz sonuçlarında her bir makine öğrenmesi algoritması için karmaşıklık matrisi elde edilmiştir. Bu karmaşıklık matrislerinin sonuçları Tablo 6'da verilmiştir.

Tablo 6. TF-IDFVectorizer yöntemiyle elde edilen karmaşıklık matrisleri

Algoritmalar		Veri Duygu Durumu	Tahmin Edilen Değerler	
			Negatif	Pozitif
NB(Kernel)	**Gerçek Değerler**	Negatif	347	148
		Pozitif	96	159
RO		Negatif	443	307
		Pozitif	0	0
DVM		Negatif	400	185
		Pozitif	43	122
KA		Negatif	441	295
		Pozitif	2	12
KEYK		Negatif	355	144
		Pozitif	88	163

RapidMiner'da yapılan duygu tespiti sonucunda pozitif ve negatif veriler / cümleler elde edilmiştir. Yapılan analizler test veri seti üzerinde gerçekleştirilmiştir. Test veri setinde 443 negatif ve 307 pozitif olmak üzere 750 veri bulunmaktadır.

Duyarlılık değeri için ilk değerlendirme negatif veriler üzerinde yapılmıştır. %71.14 oranla en iyi duyarlılık değeri oranını KEYK elde etmiştir. Pozitif veriler için %85.71 oranla en iyi duyarlılık oranını Karar Ağacı elde etmiştir.

Anma değeri için ilk değerlendirme negatif veriler üzerinde yapılmıştır. %100 oranla en iyi anma değeri oranı RO elde etmiştir. Pozitif veriler için %53.09 oranla en iyi anma değeri oranını KEYK elde etmiştir. En iyi sonuçlara ulaşan algoritmalar Tablo 7'de verilmiştir.

Tablo 7. TF-IDFVectorizer yöntemiyle elde edilen sonuçlarda en iyi oranlara ulaşan algoritmalar

Veri Duygu Durumu	Duyarlılık Değeri	Anma Değeri	Doğruluk
Negatif	KEYK	RO	DVM
Pozitif	KA	KEYK	

4. Değerlendirme ve Öneriler

Teknolojideki gelişmeler, veri ve veri bilimindeki çalışmaların daha gelişmiş ortamlarda ve imkânlarla yapılmasına olanak sağlamıştır. Özellikle veri analizi son zamanlarda popülaritesini gittikçe arttırmıştır. Çalışmamızda da Türkçe ses kayıtlarının verilerini içeren veri seti üzerinde duygu tespiti ve makine öğrenmesi algoritmalarıyla analizler yapılmıştır. Çalışma sonucunda Türkçe bir duygu veri seti elde edilmiştir.

Python programlama dili ile yapılan analizde, iki farklı vektörizasyon işleminin kullanılması çalışmanın değerlendirmelerine farklı bakış açısı sağlamıştır. CountVectorizer yöntemiyle elde edilen sonuçlarda NB algoritmasının daha çok ön plana çıktığı görülmektedir. TF-IDFVectorizer yönteminde ise DVM ikili sınıflandırmadaki etkinliğini ortaya koymuştur. Türkçe verilerle yapılan analizlerde bu iki algoritmanın kullanımıyla daha etkin sonuçlar alınabileceği söylenebilir.

RapidMiner platformuyla elde edilen sonuçlara bakıldığında doğruluk oranında DVM iyi bir sonuç elde etmesine rağmen aynı durum değerlendirilen metrikler üzerinde gerçekleşmemiştir. Hem duyarlılık değeri hem de anma değeri metriklerinde KEYK algoritması öne çıkmıştır. RapidMiner'da vektörizasyon yöntemindeki eksiklikler ve ön işlem adı verilen işlemlerin yeterli olmaması sonuçları etkileyen unsurlar olmuştur. RapidMiner'da bu yapıların geliştirilmesi analiz sonuçlarının daha başarılı olmasında etkili olacaktır.

Kullanılan platformların yetkinlikleri de çalışmadaki sonuçlara etki etmiştir. Python programlama diliyle yapılan uygulamada kullanılan Google'ın Colab ortamı, RapidMiner platformuna göre daha geniş olanaklar sunmuştur. RapidMiner'a göre daha hızlı sonuçlar elde edilmiştir ve birden fazla yöntemin kullanılmasını sağlamıştır. Bu avantaj ise çalışmanın daha iyi analizini sağlamış ve sonuçlar hakkında daha uygun değerlendirmelerin yapılmasına katkı sağlamıştır.

BERT modelinde Türkçe çalışmalar için daha iyi sonuçlar elde edilebilmesi adına BERT modelinin yapısına Türkçe durak kelimeler, Türkçe kelimelerin ve skor yapılarının eklenmesi ve geliştirilmesi için çalışmalar yapılabilir.

Kaynaklar

[1] Duygu - Vikipedi. https://tr.wikipedia.org/wiki/Duygu

[2] Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme Kavramları Arasındaki Fark Nedir? https://evrimagaci.org/yapay-zeka-makine-ogrenmesi-ve-derin-ogrenme-kavramlari-arasindaki-fark-nedir

[3] TUZCU S. Çevrimiçi Kullanıcı Yorumlarının Duygu Analizi ile Sınıflandırılması. Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi. 2020;1(2):1-5.

[4] Gondohanindijo J, Noersasongko E, Fanani AZ, Basuki RS. Comparison Method in Indonesian Emotion Speech Classification. In2019 International Seminar on Application for Technology of Information and Communication (iSemantic) 2019 Sep 21 (pp. 230-235). IEEE.

[5] Nezami OM, Lou PJ, Karami M. ShEMO: a large-scale validated database for Persian speech emotion detection. Language Resources and Evaluation. 2019 Mar;53(1):1-6.

[6] Matin R, Valles D. A speech emotion recognition solution-based on support vector machine for children with autism spectrum disorder to help identify human emotions. In2020 Intermountain Engineering, Technology and Computing (IETC) 2020 Oct 2 (pp. 1-6). IEEE.

[7] Mande AA, Dani S, Telang S, Shao Z. EMOTION DETECTION USING AUDIO DATA SAMPLES. International Journal of Advanced Research in Computer Science. 2019 Nov 1;10(6).

[8] de Pinto MG, Polignano M, Lops P, Semeraro G. Emotions understanding model from spoken language using deep neural networks and mel-frequency cepstral coefficients. In2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) 2020 May 27 (pp. 1-5). IEEE.

[9] BERT Modeli ile Türkçe Metinlerde Sınıflandırma Yapmak https://medium.com/@toprakucar/bert-modeli-ile-türkçe-metinlerde-sınıflandırma-yapmak.

[10] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. Oct 112018.

[11] Machine Learning Classification Naive Bayes https://medium.com/@ekrem.hatipoglu/machine-learning-classification-naive-bayes.

[12] Naive Bayes Classifier - Erdinç Uzun. https://erdincuzun.com/makine_ogrenmesi/naive-bayes-classifier

[13] Kirasich K, Smith T, Sadler B. Random forest vs logistic regression: binary classification for heterogeneous datasets. SMU Data Science Review. 2018;1(3):9.

[14] He LM, Kong FS, Shen ZQ. Multiclass SVM based land cover classification with multisource data. In2005 International Conference on Machine Learning and Cybernetics IEEE 2005 Aug 18 (Vol. 6, pp. 3541-3545). .

[15] Kulkarni AD, Lowe B. "Random Forest Algorithm for Land Cover Classification", International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), March 16 Volume 4 Issue 3,2016, PP: 58 - 63.

[16] Makine Öğrenimi Bölüm-4 (Destek Vektör Makineleri). https://medium.com/@k.ulgen90/makine-öğrenimi-bölüm-4-destek-vektör-makineleri.
[17] Decision Tree (Karar Ağacı): ID3 Algoritması https://erdincuzun.com/makine_ogrenmesi/decision-tree-karar-agaci-id3-algoritmasi-classification-siniflama.
[18] Makine Öğrenimi Bölüm-5 (Karar Ağaçları). https://medium.com/@k.ulgen90/makine-öğrenimi-bölüm-5-karar-ağaçları.
[19] Makine Öğrenimi Bölüm-2 (k-En Yakın Komşuluk). https://medium.com/@k.ulgen90/makine-öğrenimi-bölüm-2