www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 9, No: 4, 2022 (1303-1314)

El-Cezerî Journal of Science and Engineering
Vol: 9, No: 4, 2022 (1303-1314)
DOI : 10.31202/ecjse.1134698

ECJSE

Makale / Research Paper

Makine Öğrenmesi Algoritmaları ile Uçtan Uca Yazar Tanıma Uygulaması Geliştirme

İlayda ERDOĞAN[1], Merve GÜLLÜ[2], Hüseyin POLAT[3]

[1,2,3]Gazi Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü. Ankara/TÜRKİYE
[1]ilayerdgn@gmail.com, [2]mervegullu@gazi.edu.tr, [3]polath@gazi.edu.tr

Received/Geliş: 23.06.2022 **Accepted/Kabul:** 23.11.2022

Öz: Yüzyıllardır süregelen yazarı belirsiz metinler sorunu, internet çağının başlamasıyla oldukça artmıştır. Bu durumun en büyük sebebi internetteki verilerin çok yüksek oranını yapısal olmayan verilerin oluşturması ve bu yapısal olmayan verilerin de büyük bir bölümünü sınıflandırılmamış, yazarları belirsiz metinlerin oluşturmasıdır. Bu çalışmada makine öğrenmesi yöntemleri kullanılarak yazar tanıma problemi için web tabanlı arayüze sahip uçtan uca bir uygulama geliştirilmiştir. Metin üzerinden TF-IDF yöntemi kullanılarak öznitelikler çıkarılmış ve Destek Vektör Makineleri (DVM), Naive Bayes (NB) ve Rastgele Orman Algoritması (RO) gibi makine öğrenme algoritmaları eğitim gerçekleştirilmiştir. Test sonucunda, DVM %90 doğruluk oranıyla en iyi performansı gösteren sınıflandırıcı model olmuştur. Elde edilen DVM modeline, Python programlama dilinin kütüphanelerinden olan Flask kullanılarak bir web arayüzü geliştirilmiştir. Son olarak uygulama, kararlı ve dağıtıma uygun bir halde çalıştırılması amacıyla Docker konteynerına dönüştürülmüştür. Sonuç olarak, uçtan uca geliştirilen bir yazar tanıma uygulaması doğrudan son kullanıcı tarafından kullanılabilir biçimde sunulmuştur.

Anahtar Kelimeler: Yazar tanıma, Destek Vektör Makineleri, TF-IDF, Docker

End-to-End Authorship Identification Application Development Using Machine Learning Algorithms

Abstract The problem of unidentified texts, which has been going on for centuries, has increased considerably with the beginning of the internet age. The biggest reason for this situation is that unstructured data constitutes a very high proportion of the data on the internet, and a large part of this unstructured data is composed of unclassified texts with uncertain authors. In this study, an end-to-end application with a web-based interface was developed for the author recognition problem using machine learning methods. Features were extracted from the text using the TF-IDF method and machine learning algorithms such as Support Vector Machines (DVM), Naive Bayes (NB) and Random Forest Algorithm (RO) were trained. As a result of the test, DVM was the best performing classifier model with 90% accuracy. A web interface was developed for the obtained DVM model by using Flask, one of the libraries of the Python programming language. Finally, the application has been converted into a Docker container to run it in a stable and distribution-friendly state. As a result, an end-to-end author recognition application is made available directly to the end user.

Keywords: Author identification, Support Vector Machines, TF-IDF, Docker

Bu makaleye atıf yapmak için
Erdoğan, İ., Güllü, M., Polat, H., "Makine Öğrenmesi Algoritmaları ile Uçtan Uca Yazar Tanıma Uygulaması Geliştirme", El-Cezerî Fen ve Mühendislik Dergisi 2022, 9(4); 1303-1314.

How to cite this article
Erdoğan, İ., Güllü, M., Polat, H., "End-to-End Authorship Identification Application Development Using Machine Learning Algorithms", El-Cezerî Journal of Science and Engineering, 2022, 9(4); 1303-1314.

ORCID ID: [1]0000-0002-5749-6081; [2]0000-0001-7442-1332; [3]0000-0003-4128-2625

1. Giriş

Son zamanlarda internet teknolojisinin gelişmesi ve gittikçe artan yoğun kullanımından dolayı oluşan büyük miktarlardaki içerik verilerinin genel olarak tanımlanabilir bir yapısı yoktur. Bu tür veriler, yapısal olmayan veriler olarak adlandırılır. Yapısal olmayan bu verilerin büyük bir kısmını haberler, makaleler, araştırma bildirileri, kitaplar, sayısal kütüphaneler, e-posta iletileri ve web sayfaları gibi metinler oluşturur. Bu metinlerin analiz edilerek, içlerinden anlamlı bilgilerin çıkarılması önemli bir görev haline gelmiştir [1]. Metin analizi, metin madenciliği olarak adlandırılan bir çalışma alanının da doğmasına sebep olmuştur. Metin madenciliği, metinsel büyük veri yığınlarında gömülü olan gerçekleri ve ilişkileri tanımlar. Metin madenciliği, yapılandırılmamış metni, analize veya makine öğrenme algoritmalarına uygun normalleştirilmiş ve yapılandırılmış verilere dönüştürmek için doğal dil işlemeyi (DDİ) kullanan bir yapay zekâ teknolojisidir [2]. DDİ kullanılarak oluşturulan yapılandırılmış veriler, veri tabanlarına, veri ambarlarına veya iş zekası yapılarına entegre edilebilir ve tanımlayıcı, kuralcı veya tahmine dayalı analiz için kullanılabilir.

Hızla büyüyen internet hayatı ile yazarı belli olmayan metinlerin sayılarının fazlalığı metin yazarının tespit edilmesi problemini de ortaya çıkarmıştır. Yazar tanıma, anonim bir metin belgesinin yazarını daha önce mevcut olan metinlere dayanarak, olası adaylar kümesi içinden belirlemeye yönelik hesaplamalı bir görevdir [3]. Yazar tanıma konusu, internetin sağladığı anonimliğin çeşitli suçlar için uygun ortam oluşturmasıyla sosyal ağlarda, e-postalarda kullanıcıların yazdıkları çeşitli metinlerle insanları suistimale uğratmaları veya telif hakkı gibi problemlerin artmasından dolayı oldukça önem kazanmıştır.

Yazar tanıma alanındaki çalışmalarda bilgisayar kullanımı 1960'lara dayanır. 1970'li yıllarda dokümanları indeksleyerek sınıflandırmaya başlanan bilgisayarlı yazar tanıma çalışmalarının yoğunluk kazanması 1990'ların başını bulmuştur. Yazar tanıma görevi için günümüzde makine öğrenme algoritmaları yaygın olarak kullanılmaktadır. Makine öğrenimi, geçmiş deneyimlerden otomatik olarak öğrenme yeteneği sağlayan bir yapay zekâ teknolojisidir. Makine öğrenimi, eğitim için iyi seçilmiş girdi verileri gerektirir. Yazar tanıma görevinde yapılandırılmamış metin verilerini makine öğrenmesi algoritmalarına girdi olarak verebilmek için metin üzerinde DDİ yöntemlerini kullanarak metni temiz ve yapılandırılmış veri haline dönüştürmek gerekir. Türkçe dilindeki çalışmalarda Natural Language Toolkit (NLTK) ve Zemberek gibi kütüphaneler DDİ işlemlerinde yaygın olarak kullanılır.

Yazar tahmin çalışmalarının performansı; çalışılan dile, seçilen özniteliklere, kullanılan makine öğrenme algoritmalarına hatta işlenen metinlere göre değişim göstermektedir. Literatürde yapılmış olan çalışmaların doğruluk derecelerine bakıldığında oranların genel olarak %74 ile %97 arasında değiştiği gözlemlenmektedir [4-12].

Metinsel verilerin büyümesiyle, doğal dil işleme, makine öğrenimi ve derin öğrenme gibi yapay zeka teknolojilerinin kullanımı daha da zorunlu hale gelmektedir. Bu çalışmada, 37 yazarın köşe yazılarından oluşturulmuş 46715 metin verisi içeren bir derlem üzerinden makine öğrenmesi yöntemleri kullanılarak yazar tanıma problemi için web tabanlı arayüze sahip uçtan uca bir uygulama geliştirilmiştir. DDİ işlemleri için Zemberek ve NLTK kütüphaneleri kullanılarak veri kümesindeki metinler ön işlemden geçirildikten sonra DDİ metodolojilerinden biri olan TF-IDF (Term Frequency - Inverse Document Frequency) değerleri elde edilmiştir. Daha sonra elde edilen yapılandırılmış veriler NB (NB), Destek Vektör Makinaları (DVM) ve RO (RO) algoritmalarına uygulanmış, ve bu algoritmaların performansları incelenmiştir. En yüksek doğruluk oranını veren makine öğrenme modeli kullanılarak web tabanlı bir yazar tanıma uygulaması geliştirilmiştir. Gerçekleştirilen uygulamanın kararlılığının sağlanabilmesi ve dağıtımının kolaylaştırılması için uygulama, bir Docker konteynerına dönüştürülmüştür.

2. Materyal ve Metot

Bu çalışmada, metinlerden yazar tanıma görevini yapabilmek amacıyla yapay zeka alanındaki doğal dil işleme ve makine öğrenme yöntemleri kullanılmıştır [13-15]. Şekil 1'de makine öğrenme temelli yazar tanıma uygulamasının geliştirilme adımları gösterilmiştir. Çalışmada, 37 yazarın köşe yazılarından oluşturulmuş 46715 metin verisi içeren bir derlem kullanılmıştır. Metinler internet ortamından toplandığı için çeşitli yazım hataları ve UTF-8 standardına uymayan karakterler içermektedir. Derlem ham halde olduğu için daha sonra bu derlem üzerinde DDİ metodolojilerini kullanarak metin ön işleme, tokenize etme, normalizasyon, kelime köklerini bulma ve öznitelik çıkarımı gibi işlemler gerçekleştirilmiştir.

Derlem, Türkçe metinlerden oluştuğu için Türkçenin morfolojik yapısını başarıyla analiz edebilen Zemberek kütüphanesi ve en yaygın kullanılan DDİ kütüphanelerinden olan NLTK kütüphanesi çalışmanın önişleme evresinde kullanılmıştır. Önişleme evresinden sonra yapılandırılmamış verileri, makine öğrenmesi algoritmalarına girdi olarak beslemeye uygun hale getirmek için öznitelik çıkarım metodu olarak TF-IDF kullanılmıştır. TF- IDF'nin tercih edilmesinin sebebi, diğer yöntemlerin aksine kelimelerin sık kullanılmasına değil, bir yazarın bir kelimeyi derlemdeki diğer yazarlar daha az kullanırken incelenen yazarın daha fazla kullanmasına odaklanmasıdır.

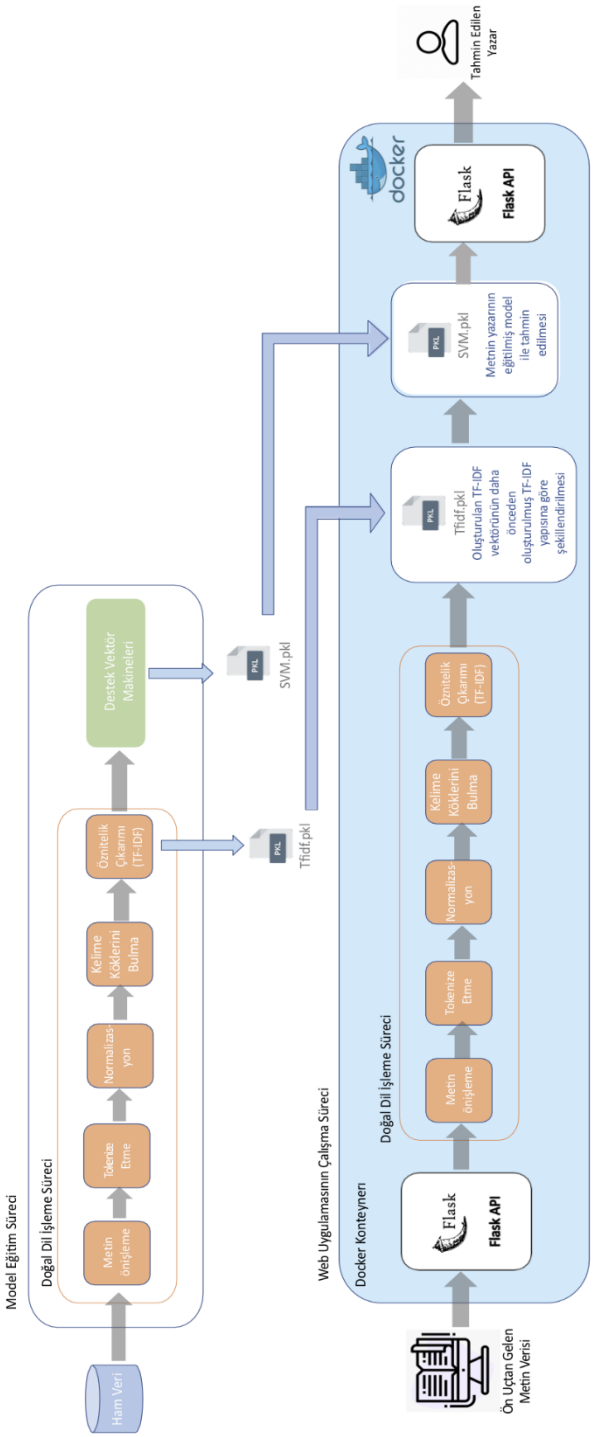
TF-IDF ile oluşturulan veri kümesi ile yazar tanıma görevini gerçekleştirmek için makine öğrenmesi algoritmalarından DVM, NB ve RO algoritmaları eğitilmiş ve test edilmiştir. Testler sonucunda elde edilen bulgulara göre %90 doğruluk oranıyla en performanslı algoritma DVM olarak tespit edilmiş ve DVM sınıflandırıcı modeli oluşturulmuştur. Daha sonra veri kümesi, DVM modeli ve Python kütüphanelerinden olan Flask kullanılarak web arayüzüne sahip bir uygulama geliştirilmiştir. Böylece DDİ metodolojileri, DVM makine öğrenme yöntemi ve Flask kütüphanesi kullanarak uçtan uca bir web tabanlı yazar tanıma uygulaması geliştirilmiştir. Son olarak geliştirilen bu uygulama, kararlı ve dağıtıma uygun bir halde çalıştırmak amacıyla Docker konteynerına dönüştürülmüştür.

2.1.1 Metin ön işleme

Metin verileri yapısal olmayan verilerdir. Yapısal olmayan verilerden anlamlı verilerin çıkarılması için verinin gürültüden / hatalardan ayıklanması, etkisiz kelimelerin ve noktalama işaretlerinin çıkarılması gerekmektedir. Temel amaç veriyi tokenizasyon işlemine hazır etmektir. Bu çalışmada da NLTK kütüphanesi yardımıyla etkisiz kelimelerin çıkarımı ve Zemberek kütüphanesi yardımıyla da dil kurallarına uymayan yanlış karakterlerin çıkarılması sağlanmıştır.

2.1.2 Tokenize etme

Metinlerdeki cümleler, kelimeler, noktalama işaretleri, sayılar, semboller gibi kavramlar DDİ literatüründe "token" olarak anılmaktadır. Tokenize etme işlemi ise işlenmeye uygun hale getirilmiş verinin cümlelere ve kelimelere parçalama işlemidir. Bu çalışmada tokenize etme işlemi Zemberek kütüphanesinin "zemberek.tokenization.TurkishTokenizer" fonksiyonuyla gerçekleştirilmiştir.



Şekil 1. Modelin geliştirilme süreci ve web uygulamasının çalışma adımları

2.1.3 Normalizasyon

Metinlerdeki yazım hatalarını tespit edip düzeltme işlemine normalizasyon denir. Bu çalışmada Zemberek kütüphanesi kullanılarak kelimeler tek tek kontrol edilip yazım yanlışı içeren kelimeler düzeltilmiştir. Bu işlem Zemberek kütüphanesinin "zemberek.normalization.TurkishSpellChecker" fonksiyonuyla gerçekleştirilmiştir.

2.1.4 Kelime kökeni bulma

DDİ adımlarından olan Kelime Kökeni Bulma, kelimelerin köklerine kadar parçalanması ve kelimelerin temel birimleri olan köklerin elde edilmesi işlemidir [38]. Bu işlemi gerçekleştirebilmek için sık kullanılan iki yöntem vardır. Bunlar "Lemmatization" ve "Stemming" olarak adlandırılır. İki yöntem de temelde aynı amaca sahip olsalar da farklı mantıklarda çalışırlar. "Stemming", kelime köklerine indirgeme işlemi sırasında sadece sondaki ekleri çıkarırken "Lemmatization", dilin morfolojik analizini baz alarak çalışır. Bu çalışmada daha bağlamsal ve daha hatasız sonuçlar vermesinden dolayı "Lemmatization" yöntemi seçilip, "zemberek.morphology.TurkishMorphology" kütüphanesi yardımıyla çalışmaya uygulanmıştır.

2.1.5 Öznitelik çıkarımı

Metin verileri, çeşitli ön işleme aşamalarından geçmiş olsalar da hala makine öğrenmesi algoritmalarına girdi olarak beslenmeye uygun hale gelmemişlerdir. Verilerin, hem daha küçük boyutlu veri tiplerine indirgenmesi hem de makine öğrenme algoritmalarının sınıflama gibi işlemler yapabilmesi için metin verilerini en uygun şekilde temsil eden değerli ve sayısal verilerin elde edilmesi gerekir. Bu işlemlerin gerçekleştirildiği DDİ adımına, öznitelik çıkarımı denir. Öznitelik çıkarımı için çeşitli yöntemler vardır [40,41].

Bu çalışmada TF-IDF vektörizasyon yöntemi kullanılarak öznitelik çıkarımı yapılmıştır. TF-IDF yöntemi temelinde Terim Frekansı (Term Frequency-TF) ve Ters Doküman Frekansı (Inverse Document Frequency-IDF) işlemlerinin birleştirilmesiyle oluşturulan bir öznitelik çıkarım metodudur. TF değeri, vektörize edilecek olan kelimenin mevcut dokümanda görülme sayısının bütün doküman derlemindeki görülme sayısına bölünmesiyle elde edilir.
TF değerinin hesaplanması eşitlik 1'de verilmiştir.

$$TF = \frac{(Terimin\ Dökümanda\ Görülme\ Sayısı)}{(Bütün\ Döküman\ Derleminde\ Terimin\ Görülme\ Sayısı)} \quad (1)$$

IDF değerinin bulunması içinse derlemdeki doküman sayısının vektörize edilecek kelimenin görüldüğü doküman sayısına bölümünün logaritması alınması gerekir. IDF değerinin hesaplanması eşitlik 2'de verilmiştir.

$$IDF = log\left(\frac{Derlemdeki\ Toplam\ Döküman\ Sayısı}{Terimin\ Görüldüğü\ Döküman\ Sayısı}\right) \quad (2)$$

TF-IDF değerinin hesaplanması da eşitlik 3'de verilmiştir.

$$T(i,j) = TF(i,j) \times IDF(i) \quad (3)$$

Eşitlik 3'de i, kelime indisini; j, doküman indisini temsil eder. TF ve IDF işlemlerinin birlikte kullanılmasının en önemli sebebi metinler arasındaki farklılıklara odaklanabilmektir. Örneğin sadece TF işlemi kullanılmış olsaydı metni sadece sık kullanılan kelimeler temsil ederdi. Dolaylı olarak bir dilde en fazla kullanılan etkisiz kelimeler birçok metni temsil ederdi. Sonuç olarak da metinleri birbirinden ayırmak zorlaşırdı. IDF değeri sayesinde de tüm derlemde sık kullanılan sözcüklerin ağırlığı azaltılmış olur. IDF işleminin en önemli faydası her dokümanda çok fazla geçen bir sözcüğün artık özel olarak bir metni temsil etme gücünün kalmamasıdır [40]. Sonuç olarak bir kelime eğer bir dokümanda fazla geçiyor fakat bütün derlemde az geçiyorsa TF-IDF'e göre o kelime bahsedilen metin için belirleyici nitelik taşıyabilecek durumdadır çünkü dokümanı temsil edecek kelimelerin dokümana özgü olması gerekir [40,42].

Çalışmada kullanılan TF-IDF vektörünün parametresi "ngram : (1,2)" olarak seçilmiştir. Bu sayede TF-IDF değerlerinin hesaplanması için alınacak kelime öbeklerinin en az ve en fazla kaç kelimeyle oluşturulacağı belirlenir [40,43].

2.2 Makine Öğrenmesi

Makine öğrenimi, geçmiş deneyimlerden otomatik olarak öğrenme yeteneği sağlayan bir yapay zeka teknolojisidir ve karmaşık sorunları yüksek doğrulukla çözmeye yardımcı olur. Bu çalışmada, yazar tanıma için veri kümesi üzerinden DVM, NB ve RO algoritmaları eğitilmiş ve test edilmiştir. Testler sonucunda DVM modeli en yüksek başarı oranını verdiği için geliştirilen uygulamada bu model kullanılmıştır.

2.2.1 Destek vektör makineleri

Destek Vektör Makineleri (DVM), 1995 yılında Vladir Vapnik, Berhard Boser ve Isabelle Guyon tarafından geliştirilmiş bir makine öğrenmesi algoritmasıdır [20,23]. İstatistiksel öğrenme yöntemlerinden olan DVM, temelde iki sınıfı birbirinden ayırmak için doğrusal bir sınıflandırıcı olarak tasarlanmıştır. DVM'nin temel amacı iki sınıfı birbirinden ayıran en uygun karar fonksiyonunun (Hiper-düzlemin) tanımlanmasıdır. En uygun karar fonksiyonu ise kendisine en yakın olan noktalar olan destek vektörleri arasındaki mesafenin maksimize edilmesiyle oluşturulur. Fakat gerçek hayattaki problemlerin her zaman doğrusal ve iki sınıflı olmayacağından dolayı DVM daha sonradan çok sınıflı ve doğrusal olmayan veriler için bir çekirdek fonksiyonu kullanarak genelleştirilmiştir. DVM'de yaygın kullanılan dört çekirdek fonksiyonu vardır. Bunlar Doğrusal, Polinominal, Sigmoid ve Radyal tabanlı fonksiyonlardır.

DVM, eğitim kümesindeki nispeten çok az veri noktasıyla bile iyi genelleme yapar, yüksek boyutlu uzaylarda etkilidir ve karar fonksiyonları için farklı çekirdek fonksiyonları kullanılabilir [24]. Sağladığı bu avantajlar nedeniyle DVM'ler literatürde oldukça yaygın kullanılan bir makine öğrenme algoritmasıdır.

2.2.2 Naive Bayes algoritması

Naive Bayes (NB) olasılık ilkelerine göre tanımlanmış bir dizi hesaplama ile bir veri örneğinin sınıfını tespit etmeyi amaçlayan denetimli öğrenme (supervised learning) algoritmasıdır. Naive Bayes, en sade hali ile bir olayın herhangi bir sınıfa ait olma olasılığını hesaplamada kullanılır. [16]. Bir veri kümesinde $c_1,c_2,\ldots,c_i$ sınıfları bulunuyor ve bir X veri örneği $x_1, x_2,\ldots,x_n$ özniteliklerinden oluşuyorsa ve olasılıkları sıfırdan farklı ise amaç C sınıfını tahmin etmektir.

$C=c_1,c_2,\ldots,c_i$ $X= x_1, x_2, ..., x_n$

Özellikle, P(C | X)'yi maksimize eden C sınıf değeri bulunmalıdır. Bayes teoremini kullanarak bütün C sınıfları için P(C | X) koşullu-sonsal olasılığı hesaplanmalıdır.

$$P(C_i \mid X) = \frac{P(C_i)\,P(C_i)}{P(X)} \tag{4}$$

P(X) olasılığı bütün sınıflar için ortak ve sabit olduğuna göre ihmal edilebilir.

$$P(x) = P(C_i)P(C_i) \ldots\ldots P(C_i)P(C_i) \tag{5}$$

$P(C_i|X)$: X olayı gerçekleştiğinde sınıf i'nin gerçekleşme olasılığı (sonsal-koşullu olasılık / posterior probability)
$P(X|C_i)$: Sınıf i'den bir X olayının gerçekleşme olabilirliği (likelihood)
$P(C_i)$: Sınıf i'nin gerçekleşme olasılığı (önsel-koşulsuz olasılık / prior probability)
P(X) : X olayının gerçekleşme olasılığı

Tüm sınıflar için $P(C_i|X)$ hesaplanıp X örneği en yüksek olasılığa sahip sınıfa atanabilir.

2.2.3 Rastgele orman algoritması

Rastgele Orman Algoritması (RO), 2001 yılında Leo Breiman tarafından geliştirilen denetimli bir öğrenme algoritmasıdır. Hem sınıflandırma hem de regresyon görevleri için kullanılabilir. Hiper parametre ayarı yapmadan bile, çoğu zaman büyük bir sonuç üreten, esnek, kullanımı kolay bir makine öğrenmesi algoritmasıdır.

Temelde bir çeşit karar ağacı algoritması olan RO algoritması eğitim aşamasında birden çok karar ağacını kullanmakta ve bu yapısıyla karar ağacı ormanı olarak da tanımlanabilmektedir. RO, birden fazla karar ağacını oluşturur ve daha doğru ve istikrarlı bir sınıflandırma/öngörü elde etmek için onları birleştirir. RO, karar ağaçlarının bir koleksiyonudur, ancak çeşitli karar ağaçları inşa etmek için veri kümesi örneklerini ve öznitelikleri rastgele seçer ve sonuçların ortalamasını alır. Algoritma rastgele örneklemeyi ve topluluk yöntemlerdeki tekniklerin iyileştirilmiş özelliklerini içermesi nedeniyle daha iyi genellemeler sunar ve geçerli tahminlerde bulunur.

3. Deneysel Bulgular ve Tartışma

Makine öğrenmesine dayalı yazar tanıma sürecinde ilk olarak metin ön işleme gerçekleştirilmiştir. Veriler üzerinde ön işlemlerin ardından öznitelik çıkarımı işlemi ile TF-IDF vektörleri elde edilmiştir. Bu vektörlerin görünümünün bir örneği "(0, 3794) 0,09979413502292324" gibidir. Örnekteki "0" değeri metin indisini, "3794" kelime indisini ve "0,09979413502292324" değeri TF-IDF değerini ifade eder. Yani bir kelimenin, belirtilen metindeki TF-IDF değeri gösterilmektedir.

TF-IDF vektörü, yapısı itibariyle sadece sınıflanacak metinle değil bütün metin derlemiyle ilgilenir. Aksi takdirde TF-IDF vektörü sınıflama gücü kalmaz. Bu nedenle TF-IDF vektörü "Vocabulary" adında daha önceden işlediği kelime veya kelime öbeklerini tuttuğu bir yapıya sahiptir. Bu çalışmada oluşturulan "Vocabulary" yapısında "ngram(1,2)" parametresi yardımıyla kelime öbekleri bir veya iki elemanlı seçilmiştir.

Yazar tanıma uygulamasının kullanılma aşamasında yeni metin belgelerini geçmişte değerlendirilen metin belgelerini referans alarak değerlendirebilmek için eğitim aşamasında şekillendirilmiş TF-IDF vektörüne ihtiyaç duyulur. Bu nedenle şekillendirilmiş TF-IDF vektörü ".pkl" dosya uzantısıyla makine öğrenme modelinden çıkarılır. Böylece metin ön işleme aşamaları sona ermiş olur.

Ön işleme aşaması bittikten sonra yazar tanıma görevinde en yüksek performans sergileyen makine öğrenme algoritmasını tespit edebilmek amacıyla DVM, NB ve RO algoritmaları eğitilmiş ve test edilmiştir. Test sonuçları Tablo 1'de verilmiştir. NB ve RO algoritmalarının doğruluk değeri dışındaki diğer performans metriklerinin düşük olması ve DVM'in aralarındaki en başarılı sonucu üretmesi uygulamanın geliştirilmesinde kullanılacak model olarak seçilmiştir. Ayrıca yazar bazında da DVM modeli daha başarılı olmuştur. Makine öğrenme algoritmalarının eğitimi tamamlandıktan

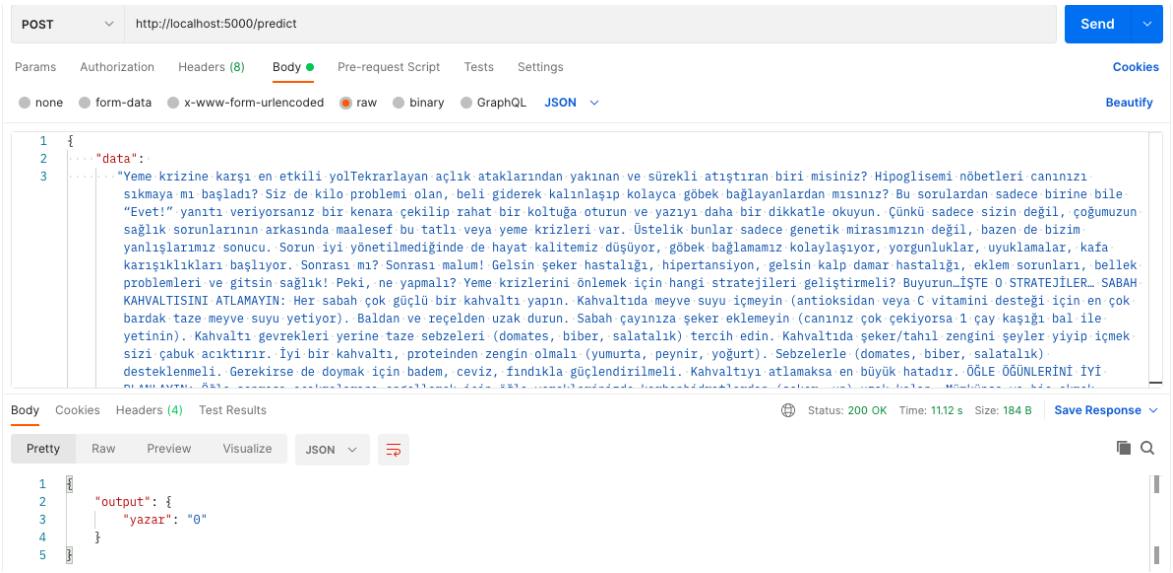
sonra en yüksek performansa sahip eğitilmiş DVM modeli de tıpkı TF-IDF vektörü gibi ".pkl" dosya uzantısıyla elde edildi.

Tablo 1. Sınıflandırma modellerin test performans değerleri

Algoritma	Doğruluk (ort)	Kesinlik (ort)	Duyarlılık (ort)	F1-Skor (ort)
DVM	0,90	0,85	0,84	0,84
NB	0,84	0,67	0,54	0,55
RO	0,80	0,78	0,61	0,65

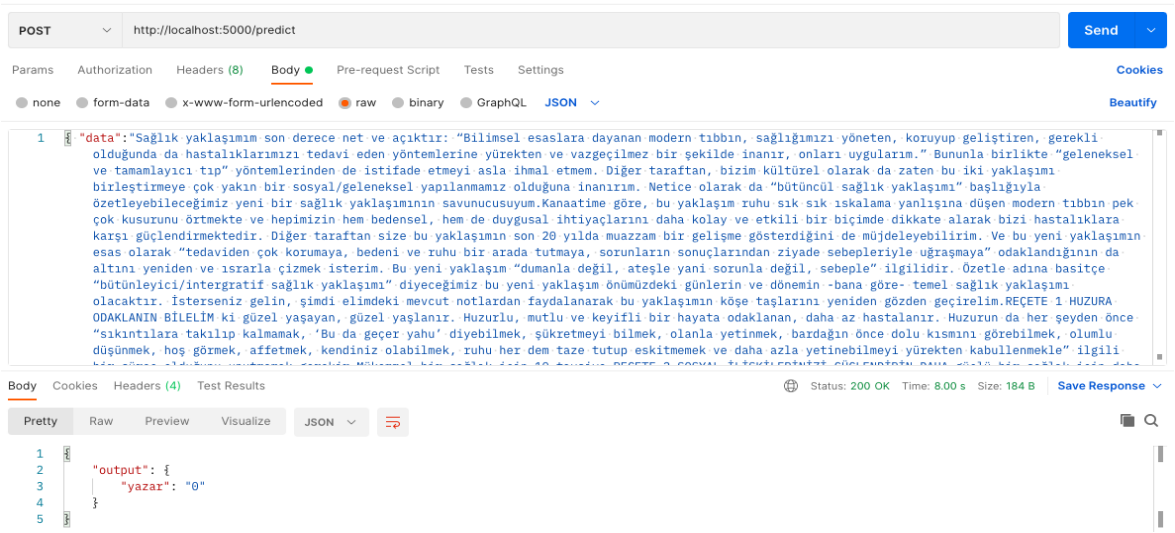
Hem TF-IDF vektörü hemde DVM model dosyalarının oluşturulması ile yazar tanıma uygulamasının web arayüzünün arka planında çalışacak gerekli dosyalar hazırlandı. Uygulamaya dışarıdan verilecek metin verilerinin hazırlık kodları da eklendikten sonra Flask kütüphanesi kullanılarak uygulamanın arka uç kısmı tamamlandı. Ayrıca uygulamanın, kararlı ve dağıtıma uygun bir halde çalıştırılması amacıyla Docker konteynerına dönüştürülmesi de gerçekleştirildi.

Daha sonra Postman uygulaması kullanılarak uygulamanın çalışma testleri yapıldı. Testler için veri kümesindeki 0 indisli yazar olan Osman Müftüoğlu'nun köşe yazısı hem de veri kümesinde olmayan, yani modelin daha önce görmediği köşe yazıları kullanılmıştır. Şekil 2'de veri kümesi içerisinde bulunan bir metin verisiyle doğru tahmin yapılmış başarılı bir Postman testinin görüntüsü verilmiştir.



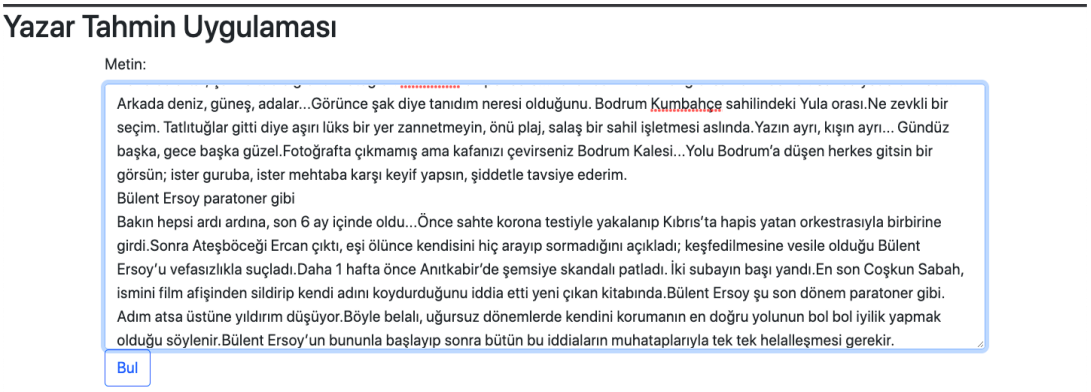
Şekil 2. Veri kümesi içerisinde bulunan bir veriyle yapılmış olan Postman testi

Şekil 3'te veri kümesi içerisinde bulunmayan bir metin verisiyle doğru tahmin yapılmış başarılı bir Postman testinin görüntüsü gösterilmiştir.



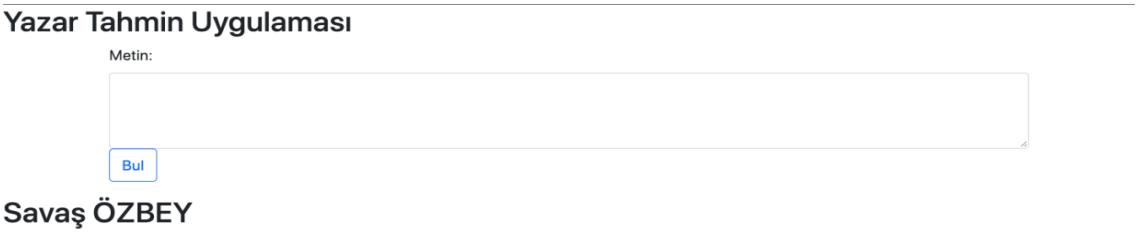
Şekil 3. Veri kümesi içerisinde bulunmayan veriyle yapılmış olan test

Şekil 2 ve Şekil 3'te açıkça görüleceği üzere "yazar:0" değerleri, uygulamaya girilmiş olan metnin 0 indisli yazarın metni olduğunu göstermektedir.
Uygulamanın arka uç kısmının testleri tamamlandıktan sonra web arayüzü oluşturulmuştur. Şekil 4 uygulamanın web arayüzü gösterilmiştir.



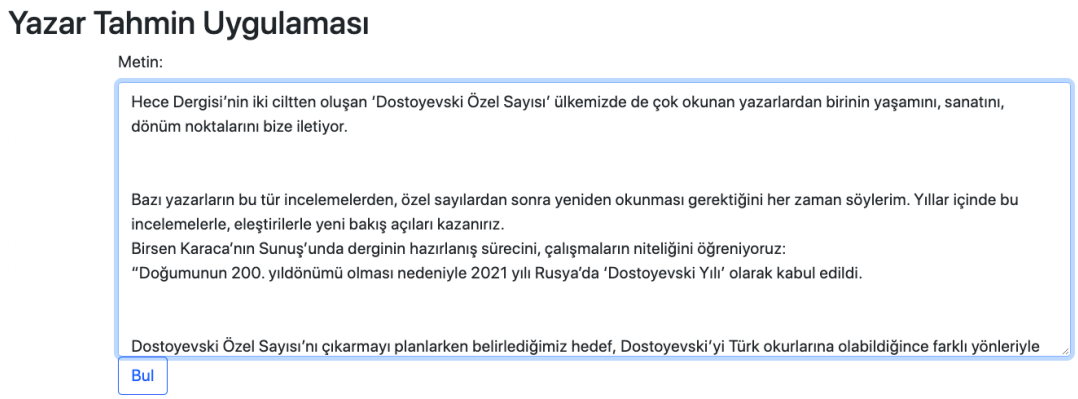
Şekil 4. Uygulama web arayüzü üzerinden Savaş Özbey'in bir metin girdisi

Şekil 4'te web arayüzü üzerinden uygulamaya modelin daha önce eğitim aşamasında hiç görmediği Savaş Özbey'in bir metni girdi olarak verilmiştir. "Bul" butonuna basıldığı anda modelin tahmini uygulamaya Şekil 5'te görüldüğü gibi yansır.

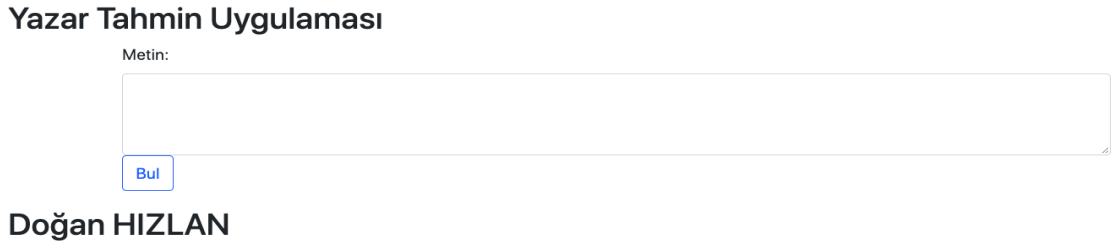


Şekil 5. Savaş Özbey'in bir metin girdisi için uygulamanın verdiği sonuç

Şekil 6 ve Şekil 7 'de, modelin daha önce görmediği Doğan Hızlan'ın bir metin verisinin girdi olarak verilmesi ve buna bağlı olarak uygulamanın çıktı olarak verdiği sonuç gösterilmiştir.



Şekil 6. Uygulama web arayüzü üzerinden Doğan Hızlan'ın bir metin girdisi



Şekil 7. Doğan Hızlan'ın bir metin girdisi için uygulamanın verdiği sonuç

4. Sonuç ve Öneriler

Bir makine öğrenme modelini doğrudan son kullanıcı tarafından kullanılabilir bir biçimde dağıtıma hazırlamak oldukça önemlidir. Bu çalışmada, uçtan uca makine öğrenmesine dayalı bir yazar tanıma uygulaması oluşturulmuştur. Geliştirilen bu uygulama, makine öğrenme çalışmalarının daha iyi sergilenmesine ve anlaşılmasına yardımcı olabilir.

Bu çalışmada öncelikle 37 yazarın köşe yazılarından oluşturulmuş toplamda 46715 metin verisi içeren bir derlem kullanılmıştır. Bu derlem, DDİ yöntemlerinden TF-IDF kullanılarak vektörize edilmiş ve bir veri kümesi oluşturulmuştur. TF-IDF vektörünün parametresi "ngram : (1,2)" olarak seçildiği için TF-IDF hesaplanması için alınacak kelime öbeklerinin en az 1 ve en fazla 2 kelimeyle oluşturulacağı belirlenmiştir. TF-IDF sadece metni incelenen yazarın diğerlerinden daha fazla kullandığı kelimelere odaklanır. N-Gram ise sadece tek kelime değil birkaç kelimeyi yan yana kullanarak bağlamdan kopmamayı sağlar. Böylece TF-IDF ve N-Gram yöntemlerinin güçlü yanları birlikte kullanılmıştır.

Başlangıçta oluşturulan derlemin DDİ yöntemleriyle işlenmesi sonucunda elde edilen veri seti kullanılarak DVM, NB ve RO makine öğrenme algoritmaları ayrı ayrı eğitilmiş ve test edilmiştir. Testler sonucunda DVM modeli %90 oranında doğruluk oranı ile NB ve RO modellerine göre daha yüksek performans göstermiştir. Yüksek başarısından dolayı DVM modeli uçtan uca yapılacak uygulamada model olarak seçilmiştir. Python kütüphanelerinden Flask kullanılarak web arayüzüne sahip bir uygulama ile model erişimi sağlanmıştır. Böylece DDİ metodolojileri, DVM makine öğrenme yöntemi ve Flask kütüphanesi kullanarak uçtan uca bir web tabanlı yazar tanıma uygulaması geliştirilmiştir. Son olarak geliştirilen bu uygulama, kararlı ve dağıtıma uygun bir halde çalıştırmak amacıyla Docker konteynerına dönüştürülmüştür.

Bu çalışmada derlemin vektörize edilmesinde TF-IDF yöntemi makine öğrenmesinde ise DVM, NB ve RO algoritmaları kullanılmıştır. Çalışmanın ilerletilmesi için, daha farklı vektörize etme yöntemleri ve makine öğrenme algoritmaları da test edilerek uygulamanın performansının

artırılmasına yönelik deneysel çalışmalar yapılabilir. Şu anki haliyle başarıyla çalışan yazar tanıma uygulamasına yazar cinsiyeti, yaşı gibi farklı özellikler eklenip uygulama güncellenebilir. Ayrıca daha geniş çaplı bir çalışma için mevcut derlem büyütülebilir. Böylece daha fazla yazarın metinleri tahmin edilebilir ve mevcut yazarlarında tahmin başarı oranları arttırılabilir. Geliştirilen web tabanlı yazar tanıma uygulaması hali hazırda basit bir arayüze sahiptir. Bu web arayüzüne yeni özellikler eklenerek görselliği geliştirilebilir.

Yazar(lar)ın Katkıları

MG ve IE çalışmada kullanılan veri kümesini birlikte oluşturdu. Veri kümesi üzerinde yapılan ön işleme ve öznitelik çıkarma işlemlerinde MG ve IE birlikte çalıştı. Öznitelikleri elde edilmiş veri kümesini kullanarak makine öğrenme algoritmalarının eğitilmesi ve test edilmesi, deneysel bulguların elde edilmesi ve daha sonrada modellerin elde edilmesinde HP ve IE birlikte çalıştılar. Oluşturulan model için web arayüzü geliştirilmesi, modelin Docker üzerinde çalıştırılması ve bu çalışmaya ilişkin dokümanın yazılmasında HP, MG ve IE ortak çalışma gerçekleştirdi. Ayrıca HP ve MG dokümanın son halini okudu ve onayladı.

Çıkar Çatışması

Yazarlar, aralarında bir çıkar çatışması olmadığını beyan eder.

Kaynaklar

[1]. Berry, M. W., “Survey of Text Mining”, Computing Reviews, 45(9),548,2004
[2]. Brocard M. L., Traore I. Saad S., Woungang I., “Authorship Verification for Short Messages using Stylometry”, Computer, Information and Telecommunication Systems (CITS), 2013
[3]. Ma J., Li Y., Teng G., Wang F. Zhao Y.,”Sequential Pattern Mining for Chinese E-mail Authorship Identification”, The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC), 2008
[4]. Diederich J., Kindermann J., Leoopold E., Paass G., “Authorship Attribution with Support Vector Machines”, Applied intelligence, 2003
[5]. Peng F., Keselj V., Cerconey N., Thomasy C., “N-gram-based Author Profiles For Authorship Attribution”, Faculty of Computing Science, Dalhousie University, 2003
[6]. Zheng R., Li J., Chen H., Huang Z., “A Framework for Authorship Identification of Online Messages: Writing Style Features and Classification Techniques”, Journal of the American Society of Information Science and Technology, 2006
[7]. Abbasi A., Hsinchun C., “Applying Authorship Analysis to Extremist-group Web Forum Messages”, IEEE Intelligent Systems, 2005
[8]. Patton J. M, Can F., “A Stylometric Analysis of Yaşar Kemal’s İnce Mehmed Tetralogy”, Computers and the Humanities, 2004
[9]. Yavanoğlu Ö., “Stilistik Özellikler Kullanılarak Yazar Tanıma İşinde Yapay Sinir Ağlarının Başarımının Değerlendirilmesi: Türkçe Köşe Yazıları”, 2017
[10]. Diri B., Amasyalı, M.F., “Automatic Turkish Text Categorization in Terms of Author, Genre and Gender”, 2006
[11]. Doğan S., “Türkçe Dökümanlar İçin N-Gram tabanlı Sınıflandırma: Yazar Tür ve Cinsiyet”, 2006
[12]. Cavnar, W. B., “Trenkle J. M., N-gram-based Text Categorization, Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Rertrieval”. Information Systems Project Management, Jolyon E. Hallows, AMACOM Pres. 1994
[13]. Kızrak, M., Bolat B., “Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma”, Bilişim Teknolojileri Dergisi, c.11, no:11 2018

[14]. [URL-1]https://www.oracle.com/tr/data-science/machine-learning/what-is-machine-learning/, 2021
[15]. [URL-2]https://www.expert.ai/blog/machine-learning-definition/#:~:text=Machine%20learning%20is%20an%20application,use%20it%20learn%20for%20themselves, 2021
[16]. [URL-3] https://www.datasciencearth.com/algorithm-naive-bayes-classifier/, 2021
[17]. Ron K., Scaling Up the Accuracy of NB Classifiers: a Decision Tree Hybrid, 2011
[18]. Rısh, I., "An Emprical Study of the Naive Bayes". IBM Research Report, 2001
[19]. Zhang H., "The Optimality of Naive Bayes", In Flaırs Conference, 2004
[20]. Vapnik, V.N., "The Nature of Statictical Learn Theory", Springer-Verlag, 1995
[21]. Kavzoğlu T., Çölkesen İ., "Destek Vektör Makineleri İle Uydu Görüntülerinin Sınıflandırılmasında Kernel Fonksiyonlarının İncelenmesi", 2010
[22]. Haykın S., "Neural Networks: A Comprehensive Foundation" ,1999
[23]. Akpınar H., DATA-Veri Madenciliği Veri Analizi, 2014
[24]. [URL-4]https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-4-destek-vekt%C3%B6r-makineleri-2f8010824054 , 2022
[25]. Shilton A., Palaniswami M., Ralph D., Tsoi A. C., "Incremental Training of Support Vector Machines", 2005
[26]. Osuna E. E., Freund R., Girosi F., "Support Vector Machines: Training and Aplications", 1997
[27]. Demirci D. A., "Destek Vektör Makineleri ile Karakter Tanıma", 2007
[28]. Cortes C., Vapnik V., "Support Vector Network", 1995
[29]. Kecman V., "Learning and Soft Computing: Support Vector Machines", 2003
[30]. Yakut E., Elmas B., Yavuz S., "Yapay Sinir Ağları ve Destek Vektör Makineleri Yöntemleriyle Borsa Endeksi Tahmini", 2014
[31]. Metlek S., Kayaalp K., "Destek Vektör Makineleri", 2020
[32]. [URL-5] https://bilgisayarkavramlari.com/2008/12/01/cok-sinifli-dvm-multiclass-svm/ ,2022
[33]. [URL-6] https://yigitsener.medium.com/destek-vekt%C3%B6r-makineleri-support-vector-machine-svm-%C3%A7al%C4%B1%C5%9Fma-mant%C4%B1%C4%9F%C4%B1-ve-python-uygulamas%C4%B1-992163ff3eec, 2022
[34]. Korkem E., "Mikroarray Gen Ekspresyon Veri Setlerinde Random Forest ve NB Sınıflama Yöntemleri Yaklaşımı", 2013
[35]. Coşgun E., Karabulut E., Karaağaoğlu E., "Random Forest ve Destek Vektör Makinası Yöntemleri ile Gen Seçimi ve Sınıflaması", 2009
[36]. [URL-7] http://zembereknlp.blogspot.com/ , 2022
[37]. [URL-8] https://www.veribilimiokulu.com/dogal-dili-anlamak-chatbot-nasil-anlar/ ,2022
[38]. Hotho A., Nürnberger A., Paab G., "A Brief Survey of Text Mining", 2005
[39]. Kaya S., "Doğal Dil İşleme Teknikleriyle Yazar-Kitap Tanıma", 2018
[40]. Aksoy N., "Türkçe Dilinde Yapılmış Açık Uçlu Sınavların Doğal Dil İşleme ile Otomatik Olarak Değerlendirilmesi", 2021
[41]. Khalid S., Khalil T., Nasreen S., "A Survey of Feauture Selection and Feature Extraction Techniques in Machine Learning" , Proceedings of 2014 Science and Information Conference, 2014
[42]. [URL-9] https://www.btkakademi.gov.tr/portal/course/dogal-dil-islemeye-giris-11864
[43]. Scikit_Learn,https://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html, 2022
[44]. [URL-10] https://web.yemreak.com/web-teknolojileri/flask , 2022
[45]. [URL-11] https://medium.com/kodlayan-nesil/flask-nedir-9364c1bb5f41, 2022
[46]. Aydemir E., Işık M., Tuncer T., "Türkçe Haber Metinlerinin Çok Terimli NB Algoritması Kullanılarak Sınıflandırılması", 2021
[47]. [URL-12]https://medium.com/deep-learning-turkiye/regresyon-ve-s%C4%B1n%C4%B1fland%C4%B1rmada-hata-metrikleri-143a40c6b656,2022