T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ
(%30 İNGİLİZCE) BİLİM DALI

METİNSEL VERİLER İÇİN ÇOK SINIFLI PROBLEMLERE HATA DÜZELTEN KOD TABANLI KOLEKTİF ÖĞRENME YÖNTEMİNİN UYGULANMASI

YÜKSEK LİSANS TEZİ

Vildan MERCAN

İstanbul
Mart – 2023

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ
(%30 İNGİLİZCE) BİLİM DALI

METİNSEL VERİLER İÇİN ÇOK SINIFLI PROBLEMLERE HATA DÜZELTEN KOD TABANLI KOLEKTİF ÖĞRENME YÖNTEMİNİN UYGULANMASI

YÜKSEK LİSANS TEZİ

Vildan MERCAN

Tez Danışmanı
Dr. Sümeyra BEDİR

İstanbul
Mart - 2023

TEZ ONAYI

Lisansüstü Eğitim Enstitüsü Müdürlüğüne;

Bu çalışma jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği (%30 İngilizce) Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman Dr. Öğr. Üyesi Sümeyra BEDİR

Üye Dr. Öğr. Üyesi Şengül BAYRAK HAYTA

Üye Dr. Öğr. Üyesi Muhammed DAVUD

Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Doç. Dr. Erhan İÇENER

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Yüksek lisans tezi olarak hazırladığım "**Metinsel Veriler İçin Çok Sınıflı Problemlere Hata Düzelten Kod Tabanlı Kolektif Öğrenme Yönteminin Uygulanması"** adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Vildan MERCAN

ÖN SÖZ

Araştırmamdaki her aşamada bana yardımcı olan ve vizyon katan değerli tez danışmanım Dr. Öğr. Üyesi Sümeyra Bedir'e, yüksek lisans eğitimim boyunca benden desteklerini esirgemeyen eşime, anneme ve kardeşlerime teşekkürlerimi sunarım.

Vildan MERCAN

İstanbul - 2023

ÖZET

METİNSEL VERİLER İÇİN ÇOK SINIFLI PROBLEMLERE HATA DÜZELTEN KOD TABANLI KOLEKTİF ÖĞRENME YÖNTEMİNİN UYGULANMASI

Vildan MERCAN

Yüksek Lisans, Bilgisayar Bilimleri ve Mühendisliği (%30 İngilizce)

Tez Danışmanı: Dr. Öğr. Üyesi Sümeyra Bedir

Mart, 2023 -52 Sayfa

Verinin incelenmesi, anlaşılması, yorumlanması, işlenmesi ve bilgisayar tarafından hakkında karar vermeye hazır hale getirilebilmesi ile, son yıllarda üzerinde fazlaca çalışılan problemlerden biri, verinin sınıflandırılması olmuştur. Bu problem, verinin belirlenen özelliklerinin sınıflandırma algoritmaları tarafından kullanılarak önceden belirlenen sınıflardan hangisine ait olacağının, etiketlenmiş veri üzerinde eğitilen bir bilgisayar tarafından tahmin edilmesine dayanan bir denetimli öğrenme problemidir. Makine öğrenmesi üzerine yöntemler geliştirildikçe bu yöntemlerin birkaç tanesinin toplu olarak kullanılıp değerlendirilmesine yönelik kolektif öğrenme metotları da geliştirilmiştir. Her ne kadar çalışmaların ilerlemesi sonucu geri beslemeli öğrenmeye dayanan derin öğrenme algoritmaları geliştirilmiş olsa da özellikle kolektif öğrenme algoritmalarının temel performanslarının değerlendirilebilmesi için makine öğrenmesi algoritmaları üzerinde analiz edilmeleri önem arz etmektedir.

Bu çalışmada, çok sınıflı metin sınıflandırma problemleri üzerinde incelemeler yapılmıştır. İki ayrı veri seti üzerinde, hata düzelten çıktı kodları olarak bilinen kolektif öğrenme metodunun performansı, standart makine öğrenmesi algoritmalarının tek başına uygulanması sonucu elde edilen performanslar ile karşılaştırılarak değerlendirilmiştir. Verinin ve sınıfların performans üzerine etkileri tartışılmış, metodun performansını arttırma olasılığına yönelik yapılabilecek geliştirmeler ile ilgili önerilerde bulunulmuştur.

Anahtar Kelimeler: Kolektif Öğrenme, Çok-sınıflı Metin Sınıflandırma, Hata Düzelten Çıktı Kodları

ABSTRACT

APPLICATION OF ERROR CORRECTING CODES BASED ENSEMBLE LEARNING METHOD FOR MULTI-CLASS TEXT CLASSIFICATION PROBLEMS

Vildan MERCAN

Master, Computer Science and Engineering (30% English)

Thesis Advisor: Asst. Prof. Dr. Sumeyra Bedir

March, 2023 -52 Pages

The classification of data as a significant problem has been studied extensively in recent years in the context of understanding, interpreting, processing, and preparing the data for computer-based decision making. This problem is a supervised learning problem based on the computer's prediction, trained on labeled data, of which of the predefined classes a data belongs to, depending on predetermined features, using classification algorithms. As methods for machine learning have been developed, ensemble learning methods are introduced for the use and evaluation of several of these methods together. Although deep learning algorithms based on back propagation and feedback learning have been developed as a result of the progress of the work, it is important to analyze machine learning algorithms, especially to evaluate the basic performance of ensemble learning algorithms.

In this study, examinations were made on multi-class text classification problems. The performance of the ensemble method known as error-correcting output codes was evaluated by comparing it with the performance obtained from the application of standard machine learning algorithms on two different datasets. The effects of the data and classes on performance were discussed, and suggestions were made for improvements to the method's performance.

Keywords: Ensemble Learning, Multi-class Text Classification, Error Correcting Output Codes

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

KISALTMALAR

LR : Lojistik Regresyon (Logistic Regression)

NB : Saf Bayes (Naive Bayes)

SVM : Destek Vektör Makinesi (Support Vector Machine)

RF : Rastgele Orman (Random Forest)

KNN : K-En Yakın Komşuluk (K Nearest Neighbor)

AdaBoost : Uyarlanabilir Güçlendirme (Adaptive Boosting)

GBDT : Gradyan Arttırma Karar Ağaçları (Gradient Boosting Decision Trees)

XgBoost : Aşırı Gradyan Arttırma (Extreme Gradient Boosting)

CatBoost : Kategorik Yükseltme (Categorical Boosting)

RASE : Rastgele Altuzaylar Topluluk Sınıflandırma (Random Subspace Ensemble Classification)

ET : İleri Derece Rastgeleleştirilmiş Ağaçlar (Extremely Randomized Trees)

MOE : Yerel Modellerin Birleşimi (Mixture of Experts)

PCA : Temel Bileşen Analizi (Principal Component Analysis)

TF-IDF : Terim Frekansı- Ters Belge Frekansı (Term Frequency-Inverse document frequency)

ECOC : Hata Düzelten Çıkış Kodları (Error Correcting Output Codes)

HD : Hamming Mesafesi (Hamming Distance)

OVO : Bire Karşı Bir (One vs One)

OVA : Bire Karşı Kalan (One vs All)

TP : Doğru Pozitif (True Positive)

TN : Doğru Negatif (True Negative)

FP : Yanlış Pozitif (False Positive)

FN : Yanlış Negatif (False Negative)

AC : Doğruluk (Accuracy)

Pr : Kesinlik (Precision)

BİRİNCİ BÖLÜM

GİRİŞ

Makine öğrenmesi, insan beyninden esinlenerek geliştirilen yapay zekaya ait bir disiplindir. Öncelikli amacı veriler üzerinden öğrenmektir. Makine öğrenmesi verilere ait kalıpları keşfeder, edindiği deneyimler üzerinden kendini eğiterek tahmin yeteneğini geliştirir. Makine öğrenmesinin temel problemlerinden birisi sınıflandırma problemidir. Sınıflandırma problemi, metin verisi veya görüntü verisi üzerinde, veri elemanlarının belirlenen sınıflara ait olup olmadığının, verisetinin önceden etiketlenmiş bir kısmı üzerinde eğitilmiş model tarafından tahmin edilmesine dayanır. Bir sınıflandırma problemi iki sınıftan birine ait olup olmadığının tahminine dayanabileceği gibi çok sınıflı yani ikiden fazla kategoriye aidiyetin kararının verilmesine dair de olabilir.

Makine öğrenmesi algoritmaları temelde ikili sınıflandırma problemleri için tasarlanmış, daha sonra çoklu sınıflandırıcılara genişletilmiştir. Hata düzelten kod tabanlı kolektif öğrenme algoritması birden fazla modelin bir arada değerlendirilmesine dayanan ve çoklu sınıflandırma problemlerini ikili alt sınıflandırma problemlerine indirgeyerek çözmeyi amaçlayan bir yöntemdir.

Makine öğrenmesi alanında araştırmalar geliştikçe günümüzün yapay zekâ teknolojilerinin kaynağını oluşturan ve geri beslemeli yeniden öğrenmeye dayanan derin öğrenme algoritmaları geliştirilmiştir. Kolektif öğrenme algoritmalarının performansının değerlendirilebilmesi için derin öğrenme algoritmalarının zaman ve donanım olarak maliyetli olması ve de verinin hacmi arttıkça kıyas için kullanılabilirliğinin pratik olmaması nedeniyle öncelikle temel makine öğrenmesi modelleri üzerinde uygulanması tercih edilmektedir.

1.1.Problem

Bu tezde Kaggle çevrimiçi veriseti paylaşım platformundan elde edilen iki ayrı metin veriseti üzerinde çok sınıflı sınıflandırma problemleri ele alınmıştır. Bu problemlerden ilki dijital sosyal medya platformlarından Twitter üzerinden elde edilen metinlerin (tweet), nefret içerikli, saldırgan veya normal olmak üzere üç sınıftan birine ait olmasının tahmin edilmesine dayanır (Mercan vd., 2021).

Diğer problem ise yine sosyal medya platformlarından YouTube üzerinden Matematik ve Bilgisayar Bilimleriyle ilgili ders videolarından alıntılanan altyazıların içeriğine göre dersin Lineer Cebir, Hesaplama (Calculus), Olasılık, Bilgisayar Bilimleri, Algoritmalar, Diferansiyel Denklemler, Yapay Zeka, Mühendislik için Matematik, Veri Yapıları, Statik ve Doğal Dil İşleme isimli on bir adet sınıftan birine ait olmasının tahmin edilmesi problemidir. Bu verisetleri üzerinde hata düzelten kod tabanlı kolektif öğrenme metodunun temel makine öğrenmesi modelleriyle birlikte performansının değerlendirilmesi tezin temel problemini teşkil etmektedir.

1.2.Amaç

Bu çalışma dijital platformlarda üretilen metinlerin sınıflandırılması problemleri üzerinde klasik makine öğrenimi yöntemleriyle ilgili bir performans kıyası sağlarken hata düzelten çıktı kodlarının sınıflandırmadaki etkisini araştırmayı ve etkinliğinin diğer yöntemlerle kıyaslanmasını; ayrıca, metin sınıflandırma problemindeki sınıf sayısının veya hata düzelten çıktı kodları uygulamasındaki parametrelerin değiştirilmesinin ECOC performansına etkileri üzerine bir değerlendirmeye ulaşmayı amaçlar.

1.3.Araştırmanın Önemi

Tezin önemi ve özgün değeri, çok sınıflı sınıflandırma problemlerini modellemenin etkili bir yolu olarak hata düzelten çıktı kodları yönteminin uygulamalarını sunması ve etkinliğini bazı makine öğrenimi teknikleriyle kıyaslamasıdır.

1.4.Temel Tanımlar ve Ön Bilgiler

1.4.1. Denetimli Öğrenme

Denetimli öğrenmede, çeşitli değişkenlerin (x) verilen değerlerine dayalı olarak çıkış değeri (y) tahmin etmek için bir model oluşturulur. Etiketli veri kümesi üzerinde verilerin sahip olduğu özelliklerle, çıkış değeri arasındaki ilişkiye en uygun fonksiyon bulunur. Eğitim verilerinden türetilen bu fonksiyon sonuç üretmek için test verilerine uygulanır.

1.4.1.1.Lojistik Regresyon (Logistic Regression)

Lojistik regresyon, sonuç değişkenlerinin bağımsız değişkenlerle ilişkisini ikili veya çoklu evrelerde olasılık olarak belirleyen bir yöntemdir. LR analizi, ortaya çıkan problemlerin verilerine göre alternatif çözümler ürettiği için birçok alanda kullanılır.

LR, belirli bir örneğin belirli bir sınıfa ait olma olasılığını hesaplayarak verileri sınıflandırmak için kullanır. Çok sınıflı sınıflandırmayı birkaç ikili sınıflandırma problemi olarak ele alır. Sonrasında birden çok sınıfa kolayca genelleştirebilir.

Modellenen ikili değişken genellikle yanıt değişkeni veya bağımlı değişken olarak adlandırılır. Yanıt ikilidir. (0,1) Sınıflandırma yaparken, 0 veya 1 fonksiyon çıkışı iki sınıfı temsil eder (Lewis, 2019).

LR'da amaç, bağımlı (yanıt değişkeni) ve bağımsız değişkenler arasındaki ilişkiyi en az değişkeni kullanarak en uygun şekilde belirleyen bir model sunmaktır. Sınıfa ilişkin karar, olasılık tahmininin bir eşikle karşılaştırılmasına veya hangi kararın beklenen optimum etkinliği sağladığını hesaplamaya dayalı olarak belirlenir (Hoi vd., 2006).

Sınıflandırma için, (x) test örneği verildiğinde, LR örneğe (y) sınıf etiketi atamanın koşullu olasılığını **(1.1)** ile doğrudan tahmin edebilir.

Burada α model parametresidir.

$$P(y|x) = \frac{1}{1+\exp\left(-y\alpha^{T}x\right)} \qquad \textbf{(1.1)}$$

1.4.1.2.Saf Bayes (Naive Bayes)

NB sınıflandırıcıları istatistiksel sınıflandırıcılardır. Belirli bir örneğin bir sınıfa ait olma olasılığı gibi sınıf üyeliği olasılıklarını tahmin ederler. NB sınıflandırıcısı Bayes Teoremine ve öznitelik bağımsızlığı varsayımına dayanmaktadır (Webb, 2017).

Bir öznitelik değerinin belirli bir sınıf üzerindeki etkisinin diğer öznitelik değerlerinden bağımsız olduğunu varsayar. Bu varsayıma koşullu bağımsızlık varsayımı denir.

NB sınıflandırıcıları, ayarlanması gereken serbest parametrelere sahip değildir. Bu, tasarım sürecini büyük ölçüde basitleştirir. Ayrıca, sınıflandırıcının olasılık değerleriyle çalışması, bu sonuçları çok çeşitli görevlere uygulamayı, rastgele bir ölçeğin kullanılmasından daha kolay hale getirir. NB sınıflandırıcıları, öğrenme

başlamadan önce büyük miktarda veri gerektirmez. Sınıflandırıcıları karar verirken hesaplama açısından hızlıdır (Stern vd., 1999).

NB hesaplanırken belgede ilgili özellik varsa C=1 değerini atanır, aksi takdirde 0 değeri atanır. Metin sınıflandırmasında öznitelik seçiminden sonra, NB sınıflandırıcısı, i'nin sunduğu tüm belgelerden oluşan metin alt uzayını her i'ye göre bölümlere ayırır. NB sınıflandırıcısı belgeyi en yüksek olasılığa sahip sınıfa atar.

$D = \langle di \rangle$, $(i = 1, \dots, n)$ sınıflandırılacak problem örneğinin vektörü, $C = \{c_1, c_2, \dots, c_k\}$ tanımlanmış sınıflar, $P(c_j)$ c_j sınıfının görülme olasılığı, $(j = 1, .., k)$ $P(d_i|c_j)$, D belgesinin sınıflar uzayında dağılım olasılığı, $C^*(x)$ sınıf etiketi atayan fonksiyon olmak üzere Naive Bayes formülü **(1.2)** deki gibidir.

$$C^*(D) = \arg\max P(c_j) \prod_i P(d_i|c_j) \qquad \textbf{(1.2)}$$

1.4.1.3.Destek Vektör Makinesi (Support Vector Machine)

SVM, istatistiksel öğrenme teorisine dayanmaktadır. Konuşma tanıma, metin sınıflandırma gibi işlemlerde veri ayrımı için kullanılan güçlü bir araçtır.

SVM'ler, bir dizi girdi değişkeni ile sistemin çıktısı arasında mevcut olan bilinmeyen ilişkiyi üretme potansiyeline sahiptir. Etiketli eğitim verisi örneğine göre oluşturduğu modelle, yeni örneğin kategorisini tahmin eder.

SVM, ikinci dereceden optimizasyon problemi çözülerek eğitilir. SVM'nin tercih edilme sebeplerinden biri de model karmaşıklığı ve tahmin hatasını aynı anda indirebilmesidir (Jakkula, 2016).

SVM'ler, öznitelik uzayında sınıflandırma problemi daha basit hale geldiği için veri kümelerinin girdi vektörünü yüksek boyutlu bir öznitelik uzayına eşler.

Bu uzayda iki sınıf arasında optimal bir ayırıcı olarak hiper düzlem kullanılır. Hiperdüzlem yardımıyla genelleme hatası en aza indirilirken marj en üst düzeye çıkarılır. Buradaki marj, ayırıcı hiper düzlem ile iki sınıfın her iki tarafındaki en yakın noktalar arasındaki mesafenin bir toplamı olarak hesaplanır.

Ayırıcı hiper düzlem, iki paralel hiper düzlem arasındaki mesafeyi maksimize eden hiper düzlemdir. SVM'nin amacı, farklı sınıflara ait veri noktalarını ayıran özellik uzayında karar sınırları oluşturmaktır (Parveen & Singh, 2015).

En basit haliyle, SVM'ler, eğitim verilerini maksimum bir marjla ayıran hiper düzlemlerdir. Hiper düzlemin bir tarafında bulunan tüm vektörler -1 olarak etiketlenir ve diğer tarafta bulunan tüm vektörler 1 olarak etiketlenir. Hiper düzleme en yakın olan eğitim örneklerine destek vektörleri denir.

$\{x_1, x_2, .. x_n\}$ x_i giriş modelleri, $y_i \in \{-1, 1\}$ olmak üzere $\{y_1, y_2, \dots, y_n\}$ etiketlenmiş verileri, ω ağırlık vektörü, b sabit, M maksimum marjı gösterir. Sınıflar +1, -1 olarak tanımlandığından, sınıfları bölen çizginin denklemi **(1.3)** gibidir. Maksimum marj **(1.4)** deki gibi hesaplanır.

$$y = \sum_{i=1}^{n} \omega_i x_i + b = x_i \omega + b \quad \textbf{(1.3)}$$

$$M = \frac{2}{\|\omega\|} \quad \textbf{(1.4)}$$

1.4.1.4.Rastgele Orman (Random Forest)

RF Algoritması, bir karar destek tekniği olan karar ağaçları koleksiyonundan oluşur. Karar ağaçları, tahmin yapmak için eğitim verilerine bağlı olarak farklı çıktılar üretir.

Topluluktaki her ağaç, önyükleme örneği adı verilen eğitim kümesinin değiştirmeli alt kümelerinden oluşturulur ve nihai çıktı, ortalama veya çoğunluk sıralamasına dayanır.

RF'daki, karar ağaçlarının her biri bağımsız olarak örneklenen rastgele bir vektörün değerlerine bağlıdır ve tüm ağaçlar için aynı dağılıma sahiptir. Karar ağaçları yaprak düğümlerine ve yapraklara sahip aşağıya doğru uzayan bir yapıdadır. Bu ağaçlar önceden belirlenmiş bir durdurma koşulu sağlanana kadar verilen veri kümesini tekrar tekrar ikili gruplara ayırır. Bölme ve durdurma kriterlerinin ayarlanma şekline bağlı olarak hem sınıflandırma görevleri (kategorik sonuç) hem de regresyon görevleri için tasarlanabilir (Koulis, 2003).

H(x) sınıflandırıcı fonksiyon (sınıflandırma modellerinin toplamı), h_i tek bir karar ağacının temsili, Y hedef değişken (sınıflandırma etiketi), I(*) gösterge fonksiyonu olmak üzere, rastgele ormanın karar verme formülü (1.5) deki gibidir.

$$H(x) = \arg\max \sum_{i=1}^{k} I(h_i(x) = Y) \quad \textbf{(1.5)}$$

1.4.1.5.K-En Yakın Komşuluk (K Nearest Neighbor)

KNN yöntemi, uygulama kolaylığı ve performansı sebebiyle yaygın olarak kullanılan makine öğrenimi modelidir. KNN etiketlenmemiş her örneği, eğitim setindeki en yakın k komşusu arasındaki çoğunluk etiketine göre sınıflandırır. Bu nedenle performansı, en yakın komşuları belirlemek için kullanılan mesafe metriğine önemli bağlıdır(Sun & Huang, 2010).

KNN son derece esnek bir sınıflandırma şemasıdır ve eğitim verilerinin herhangi bir ön işlemesini içermez. Bu çok büyük problemlerde hem yer hem de hız avantajı sunabilir (Jiang vd., 2012).

Ön bilginin yokluğunda, çoğu KNN sınıflandırıcısı, vektör girdileri olarak temsil edilen örnekler arasındaki farklılıkları ölçmek için basit Öklid metriğini kullanır.

X bir örneği tanımlayan vektör, N örneğe ait öznitelik sayısı (vektörün boyutu), ω_r: r. niteliğin ağırlığı, A_r: r. Örneğin özelliği ve $(1 < r < n)$ olmak üzere,

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} \omega_r(\alpha_r(x_i) - \alpha_r(x_j))^2} \quad \textbf{(1.6)}$$

x_i ve x_j arasındaki mesafenin azlığı ölçüsünde birbirine benzerdir.

d_i: bir test örneği, x_j : eğitim kümesindeki k en yakın komşularından biri, $y(x_j, c_k)$ x_j nin c_k sınıfına ait olup olmadığını gösterir.

Bir test örneğine atanan sınıf etiketi, k en yakın komşusunun çoğunluk oyu ile belirlenir. Tahmin sonucu, k en yakın komşuda en çok üyeye sahip olan sınıf olacaktır.

$$y(d_i) = \arg\max_k \sum_{x_j \epsilon kNN} y(x_j, c_k) \quad \textbf{(1.7)}$$

1.4.2. Denetimsiz Öğrenme

Denetimsiz öğrenme, kümeleme yaklaşımına dayanmaktadır. Denetimsiz öğrenme algoritmaları, verilerdeki yapıyı kendileri keşfetmek üzere tasarlanmıştır (Kovács & Terstyanszky, y.y.).

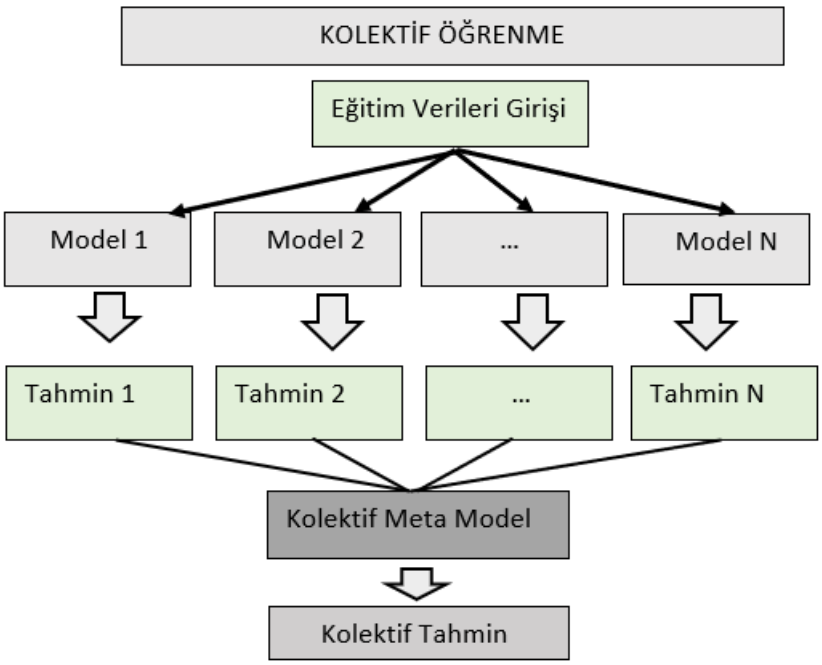
Model geliştirmek için etiketli veriler kullanılmaz. Sistem ağ oluştururken girdi-çıktı çiftlerindeki düzenlilikleri keşfederek verileri kümeler halinde düzenler.

Farklı kümeler, girdilerin farklı özelliklerini temsil eder. Algoritma verilerden kalıplar çıkarır ve etiketleri oluşturur. Yeni veriler tanıtıldığında, verilerin sınıfını tanımak için önceden öğrenilen kalıplar kullanılır. Böylece yeni veriler mevcut kümelere atanır.

İKİNCİ BÖLÜM

KOLEKTİF ÖĞRENME (ENSEMBLE LEARNING)

Kolektif öğrenme, birden fazla makine öğrenimi algoritmasını kullanmak ve bu algoritmaların tahminlerini birleştirmektir. Kolektif öğrenme yöntemleri, veriler üzerinden çıkarılan özelliklere dayalı olarak sonuçlar üretmek için çoklu makine öğrenimi algoritmalarından yararlanır. Daha iyi performans elde etmek için algoritmalardan elde ettiği sonuçları çeşitli oylama mekanizmalarıyla birleştirir (Witten vd., 2017).



Şekil 2.1: Kolektif Öğrenme Model Mimarisi

Kolektif öğrenme doğru bir tahmin elde etmek için öngörüleri birleştirir.

Birden çok modeli birleştirirken, tek bir modele ait hataların diğer modeller sayesinde telafi edilmesi hedeflenir. Böylece topluluğun genel tahmin performansı, kendisini oluşturan her bir modelden daha iyi olabilir. Kolektif yöntemler, karşılaşılan bazı zorluklara sunduğu etkili çözümler sayesinde de tercih edilmektedir.

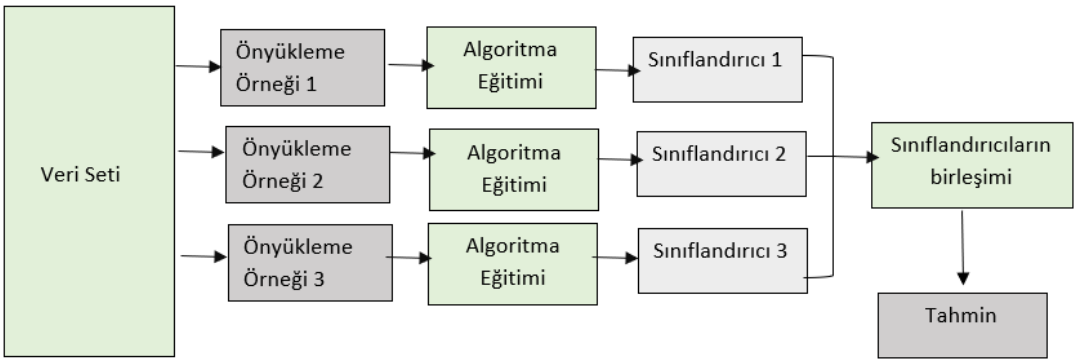
Makine öğreniminde algoritmalar azınlık sınıflarını göz ardı ederken, çoğunluk sınıfı için tercih geliştirerek sınıf dengesizliği problemlerine sebep olabilir. Kolektif yöntemler, sorunu hafifletecek şekilde uygulanabilirdir. Kolektif yöntemler, tekli makine öğrenimi modellerinin varyans ve yanlılık hatalarını sınırlayabilir (Mienye & Sun, 2022).

Gerçek zamanlı makine öğrenimi uygulamalarında, özelliklerin ve etiketlerin dağılımının zaman içinde değişimiyle kavram kayması meydana gelebilir. Tahmin performansını etkileyen bu durum topluluk tabanlı yaklaşımlarla çözülebilir.

Makine öğrenimi algoritmalarında boyutluluğun artması verilerin bulunduğu alanın hacmini arttırır. Bu sebeple yüksek boyuttaki bu uzayda veriler seyrekleşir. Boyutsallığın laneti olarak bilinen bu sorunun etkisi belirli topluluk yöntemleriyle önlenebilir (Geurts vd., 2006).

2.1.Torbalama (Bagging)

Torbalama, nesnelerin önyükleme örneklerini alır ve her örnek üzerinde bir sınıflandırıcı eğitir. Torbalama, her bir sınıflandırıcının orijinal veri kümesinden yedek olarak alınan önyükleme örneklerini kullanarak eğitildiği bağımsız bir modeller topluluğu oluşturmayı hedefleyen yaklaşımdır.



Şekil 2.2: Torbalama Model Mimarisi

Sınıflandırıcı başına yeterli sayıda örnek sağlamak için, her örnek genellikle orijinal veri kümesindekiyle aynı sayıda örnek içerir. Torbalama, sayısal bir sonucu tahmin ederken sınıflandırıcılardan elde ettiği sonuçların ortalamasını alır ve bir sınıfı tahmin ederken çoğul oylama yapar (Breiman, 1996).

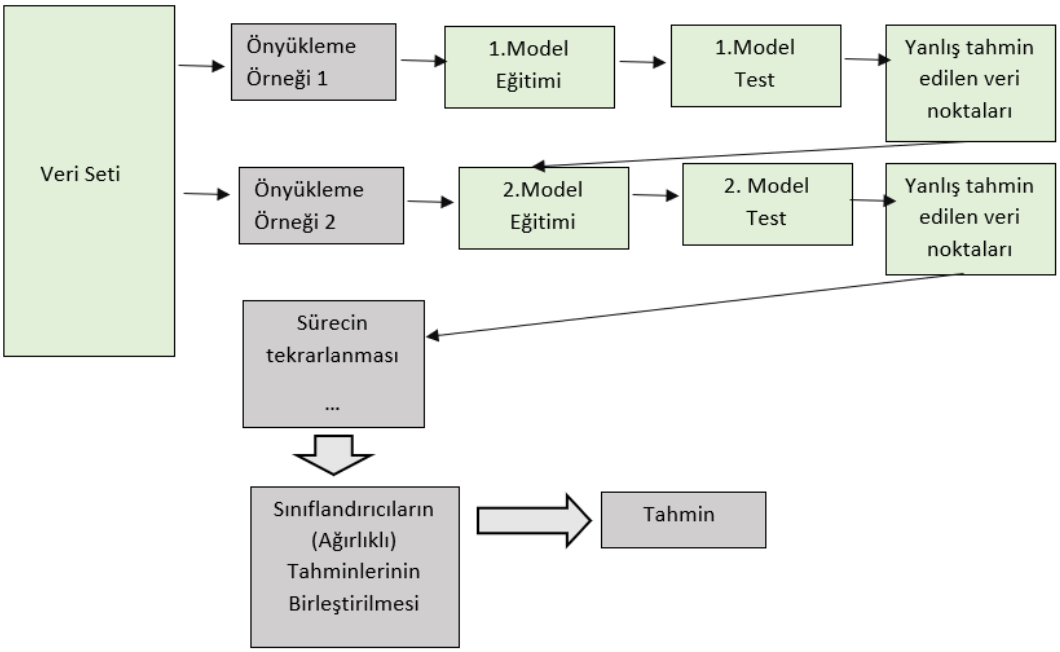
Tahmin edicileri torbalama, bir tahmin edicinin birden çok sürümünü oluşturarak bunları toplu bir tahmin edici elde etmek için kullanmaktır. Öğrenme setinin önyükleme kopyaları yapılır ve bunlar yeni öğrenme setleri olarak kullanılır. Böylece çoklu versiyonlar oluşturulmuş olur. Tahmin zamanında, temel sınıflandırıcıların çıktıları toplanır.

Torbalama, istikrarsız tahmin veya sınıflandırma şemalarını iyileştirmek için yeni ve başarılı bir yöntemdir. Torbalama performansını sağlayan etkenler, ortalama almak ve önyargıları arttırmadan bireysel sınıflandırıcıların varyansını azaltmaktır. Problemin karmaşıklığı veya ölçeği nedeniyle tek adımda iyi bir sınıflandırıcı bulmanın zor olduğu büyük veri seti problemleri için son derece yararlıdır. Torbalama, bir öngörücünün varyansını azaltmak için de etkilidir (Collell vd., 2018).

2.2.Güçlendirme (Boosting)

Sınıflandırıcı-öğrenme sistemlerinin performansını iyileştirmeye yönelik bu teknik, zayıf sınıflandırıcılardan güçlü bir sınıflandırıcı oluşturmaya çalışan bir topluluk modellemesidir. Yöntem olarak "zayıf" bir sınıflandırıcının performansını bir topluluk yapısı içinde kullanarak arttırmayı amaçlar (Rodríguez vd., 2006).

Güçlendirmede ilk model oluşturulduktan sonra, ikinci modeli eğitmek için ikinci önyüklemeli örneğe ek olarak ilk modeldeki yanlış sınıflandırılmış veri noktaları da dahil edilir. Her bir model bir öncekiyle bu ilişkiyi yineleyerek sıralı bir sistem oluşturur.



Şekil 2.3: Güçlendirme Model Mimarisi

Zayıf modelleri seri halinde kullanarak bir model oluşturulur. Topluluktaki sınıflandırıcılar ardışık olarak eklenir. İlk olarak, eğitim verilerinden bir model oluşturulur.

Veri kümesindeki nesnelerin ağırlıkları düzenlenir, böylece sınıflandırılması zor olan nesneler daha fazla ağırlık kazanır ve sonraki sınıflandırıcıların onlara odaklanması sağlanır. Daha sonra birinci modelde var olan hataları düzeltmeye çalışan ikinci model kurulur. Sonraki her sınıflandırıcı, önceki topluluk üyelerinin sınıflandırmakta zorlandığı veriler üzerinde eğitilir. Sistem tekrar tekrar çağrılır ve öğrenilen sınıflandırıcılar, genellikle herhangi bir bileşeninden daha yüksek tahmin doğruluğuna sahip olan bileşik bir sınıflandırıcıda birleştirilir.

2.3.Uyarlanabilir Güçlendirme (Adaptive Boosting)

AdaBoost, birden çok zayıf sınıflandırıcıyı birleştirerek güçlü bir sınıflandırıcı oluşturan yükseltme algoritmasıdır. Orijinal veri seti üzerinde kolay bir dille tahminler yaparak başlar ve ardından her gözleme eşit ağırlık verir. Zayıf öğrenici çıkışı diğer öğreniciye giriş olacak şekilde eğitilir.

İlk öğrenicinin tahmini yanlışsa, daha yüksek önemi yanlış tahmin edilen ifadeye vererek yinelemeli bir süreç oluşturur. Modelde sınıra ulaşılana kadar zayıf öğrenici çıkışı diğer öğreniciye giriş olacak şekilde eğitilerek yeni öğrenenler eklemeye devam edilir (Li vd., 2017).

AdaBoost'ta karar ağaçlarının derinliği 1'dir (yani 2 yaprak). AdaBoost'ta son sınıflandırmaya karar vermek için ormandaki her bir karar ağacının yaptığı tahminler nihai tahmin üzerinde farklı etkilere sahiptir.

2.4.Gradyan Arttırma Karar Ağaçları (Gradient Boosting Decision Trees)

GBDT algoritması, birden fazla modeli sırayla eğitilirken her yeni model için Dereceli alçalma (Gradient Descent) yöntemini kullanarak kayıp fonksiyonunu kademeli olarak en aza indirmeye çalışır. Algoritma sıralı olarak oluşturulur.

Model her yeni ağaç için son ağacın hatalarını dikkate alır. Model türevlenebilir kayıp fonksiyonuna uygulanan arttırmanın genelleştirmesiyle oluşturulur (Cheng vd., 2019).

Gradyan arttırma, gradyan inişini gerçekleştirmeye odaklanan bir topluluk tahmincisi oluşturma sürecidir. Veri noktalarının ağırlıklarını ayarlamak yerine tahmin ile gerçek arasındaki farka odaklanır.

Karar ağaçlarını zayıf eğimli olarak alır çünkü bir karar ağacındaki düğümler, en iyi ayrımı seçmek için farklı bir özellik dalını dikkate alır. Topluluğu oluşturan ağaçların farklı olması sayesinde verilerden her zaman farklı çıktılar yakalayabilirler (F. Zhang vd., 2016).

2.5.Aşırı Gradyan Arttırma (Extreme Gradient Boosting)

XGBoost, ağaç güçlendirme için ölçeklenebilir bir makine öğrenme sistemidir. XGBoost' da sıralı topluluk tekniği olarak da adlandırılan sıralı bir karar ağacı oluşturulur. Bu yöntemde, veri tabanındaki veri değerlerinin her birine, daha sonraki analizler için bir karar ağacı tarafından seçilme olasılığını tanımlayan bir ağırlık değeri atanır (Dhaliwal vd., 2018).

Başlangıç ağırlık değeri her veri değeri için aynıdır ve analize göre değişir. Veri değerlerinin ilk sınıflandırıcıdan geçişinin sonuçları, kendisinden bir önceki sonuçları

koruyan ve bunun üzerine inşa edilen yeni bir sınıflandırma modeli oluşturmaya yardımcı olur. Bu süreç, son sınıflandırıcı oluşana kadar devam eder.

XGBoost, yeni ağaçlar oluşturmak için sürekli özellik bölme yoluyla birçok zayıf sınıflandırıcıya uyarlanır ve zayıf sınıflandırıcıları güçlü bir öğreniciye ekler.

Kullanılan temel sınıflandırıcılar, birlikte çalışabilmeleri için doğrusal olarak üst üste bindirilir. XGBoost amaç fonksiyonunu genişletir ve algoritmanın doğruluğunu ve hızını artırır (Chen & Guestrin, 2016).

XGBoost algoritması, gradyan artırma algoritmasının benzer bir versiyonudur. Amaç fonksiyonunun yalnızca ilk türev bilgisini çıkaran GBDT algoritmasıyla karşılaştırıldığında, XGBoost, gradyan artırma çerçevesinin verimli ve ölçeklenebilir bir uygulamasıdır.

2.6.Kategorik Yükseltme (Categorical Boosting)

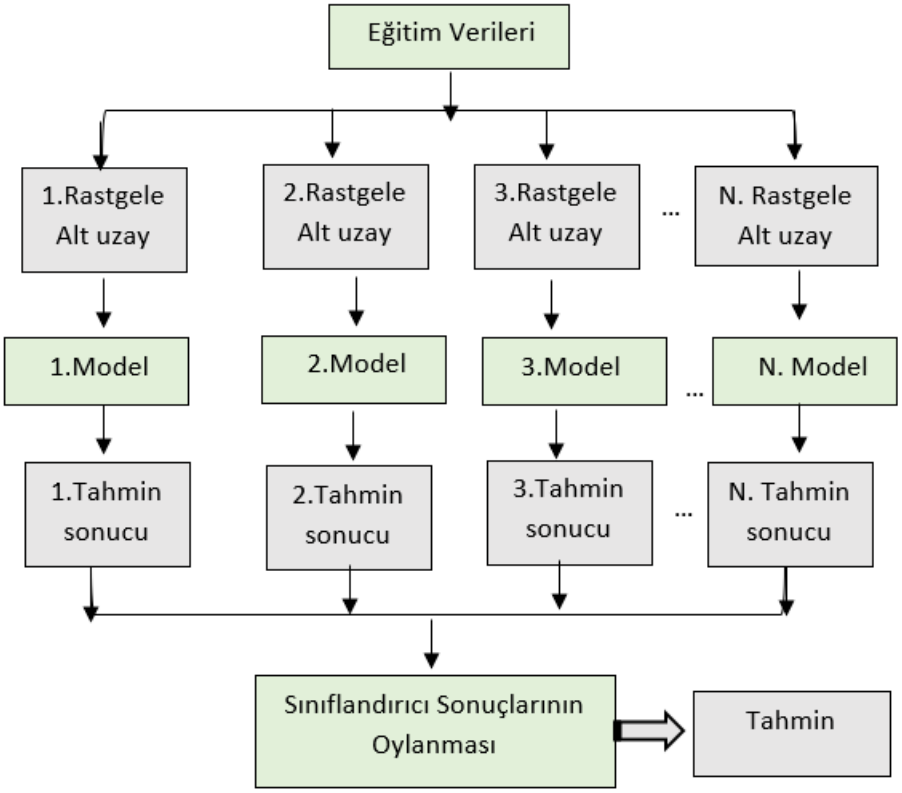
Catboost, karar ağaçları tabanlı gradyan yükseltme algoritmasıdır. CatBoost'ta, temel tahmin ediciler, karar tabloları olarak adlandırılan habersiz karar ağaçlarıdır. Habersiz terimi, ağacın tüm seviyesinde aynı bölme kriterinin kullanıldığı anlamına gelir. Bu tür ağaçlar dengelidir, aşırı uydurmaya daha az eğilimlidir. Ağacın her bir yaprak düğümünün dizini, uzunluğu ağacın derinliğine eşit olan bir ikili vektör olarak kodlanır, bu da model parametre ayarı ihtiyacını azaltır. Ağacın aynı derinliğinde aynı bölümleme kriterlerini kullanmak, karar ağacının dengeye ulaşmasını, fazla uydurmayı önlemesini ve model işleme özelliklerinin hızını arttırmasını sağlar (Prokhorenkova vd., 2018).

CatBoost ile diğer arttırma algoritmaları arasındaki farklardan biri, CatBoost'un simetrik olarak ağaçlar üretmesidir.

2.7.Rastgele Altuzaylar Topluluk Sınıflandırma (Random Subspace Ensemble Classification)

RASE, kendisini oluşturan her bir topluluk üyesini orijinal özellik kümesinden rastgele seçilen farklı bir alt küme üzerine inşa eder. Bireysel sınıflandırıcılar birbirinden bağımsız olarak oluşturulur.

Eğitim veri kümesindeki farklı özellik alt kümelerinde eğitilmiş birden çok sınıflandırıcıdan gelen tahminler birleştirilir (Kotsiantis & Kanellopoulos, 2012).

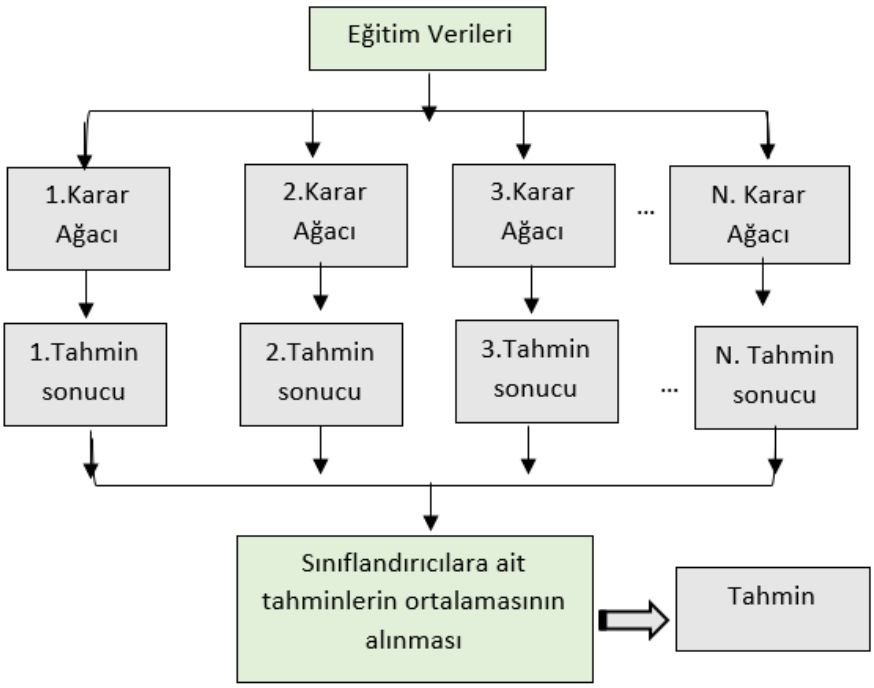


Şekil 2.4: RASE Model Mimarisi

RASE yöntemi genellikle karar ağaçlarıyla kullanılsa da performansı girdi özelliklerinin seçimiyle anlamlı bir şekilde değişen herhangi bir makine öğrenimi modeliyle de etkili şekilde kullanılabilir.

2.8.İleri Derece Rastgeleleştirilmiş Ağaçlar (Extremely Randomized Trees)

Ekstra Ağaçlar olarak da bilinen Son Derece Rastgeleleştirilmiş Ağaçlar, eğitim süresi boyunca veri kümesi üzerinde farklı özellik alt kümeleriyle birden çok ağaç oluşturur.



Şekil 2.5: ET Model Mimarisi

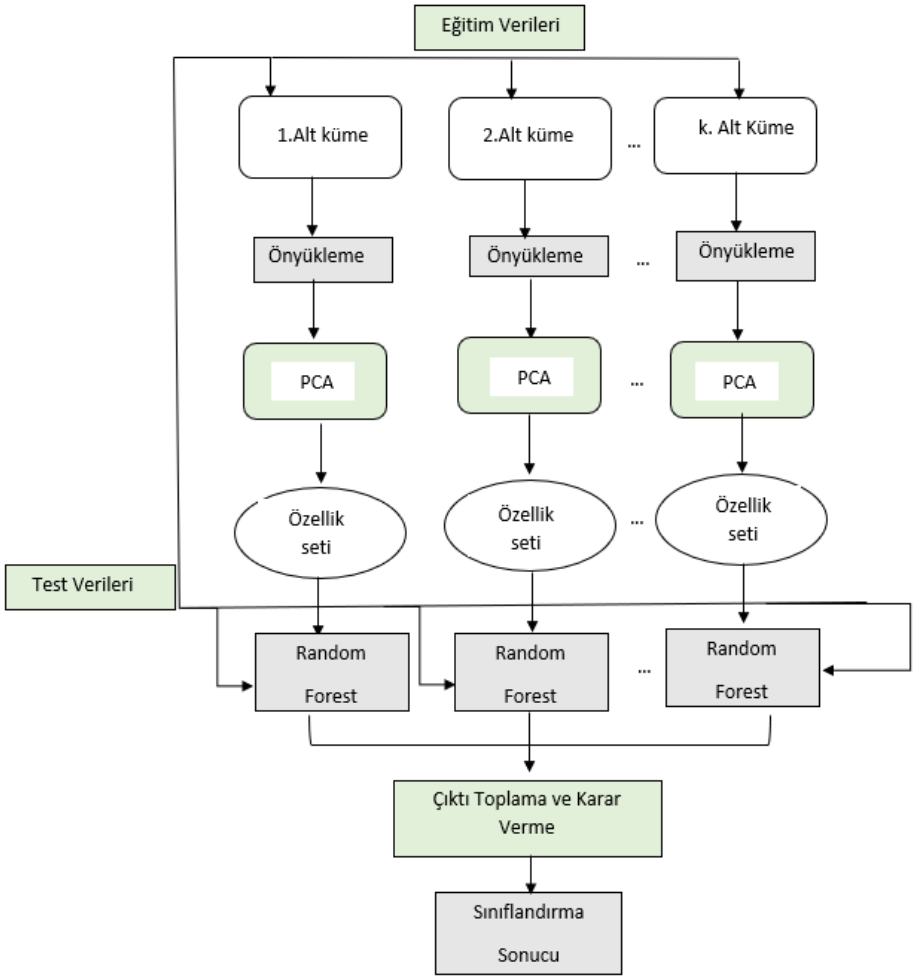
Her karar ağacı için ET algoritması ağaç düğümünü bölerken hem özniteliği hem de kesim noktası seçimini rastgele olacak şekilde seçer. Rastgele bölme işleminde uygun bir parametre seçimi yapılabilir veya işlem problemin özelliklerine ayarlanabilir (Geurts vd., 2006).

2.9.Rotasyon Ormanı (Rotation Forest)

Rotation Forest, kendisini oluşturan her ağaç için özniteliklerin alt kümelerinde dönüşümler gerçekleştiren bir topluluk öğrenmesi yöntemidir. Bir temel sınıflandırıcı için eğitim verileri oluşturulurken, özellik seti rastgele k alt kümeye böler. k, algoritmanın bir parametresidir. Bu alt kümelere PCA (Temel Bileşen Analizi) uygulanır.

Temel sınıflandırıcı için yeni öznitelikleri oluşturmak üzere k eksenli döndürmeler gerçekleşir. Karar ağaçları, özellik eksenlerinin dönüşüne duyarlıdır. Döndürme yaklaşımı fikri, topluluk içindeki bireysel doğruluğu ve çeşitliliği aynı anda destekler.

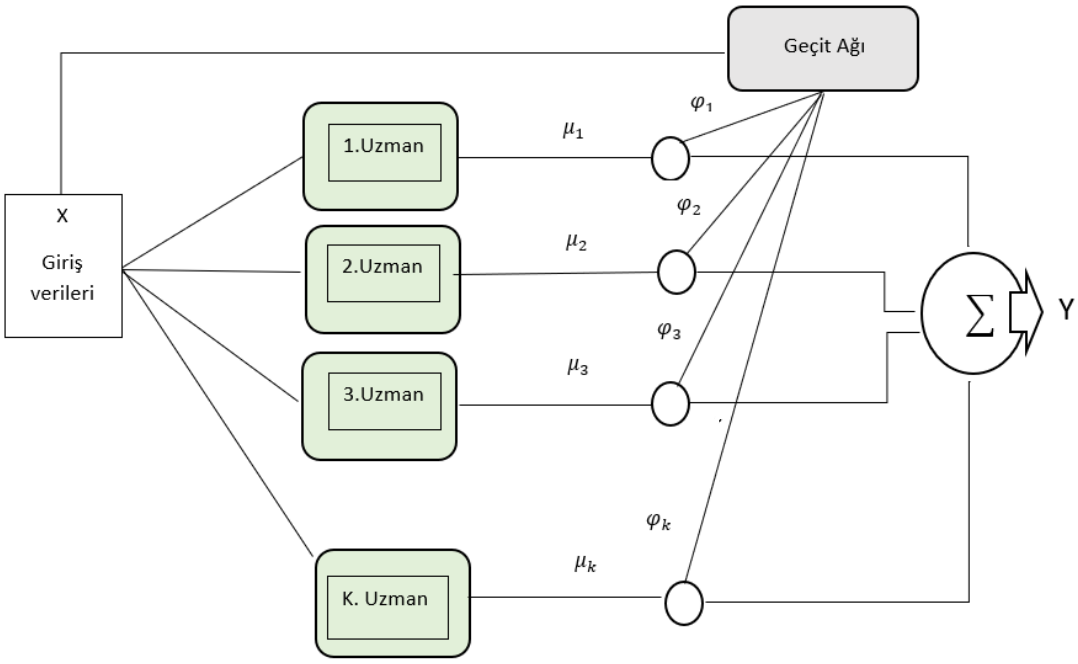
Çeşitlilik, her temel sınıflandırıcı için özellik çıkarımı yoluyla desteklenir. Doğruluk, tüm ana bileşenleri koruyarak ve her temel sınıflandırıcının eğitiminde veri setinin tamamı kullanarak sağlanır (Rodríguez vd., 2006).



Şekil 2.6: Rotasyon Ormanı Model Mimarisi

2.10. Yerel Modellerin Birleşimi (Mixture of Experts)

MoE, yerel kalıpların denetimli öğrenme vakasına genişletilmesidir. Akademik araştırmalarda ve endüstri uygulamalarında yaygın olarak kullanılan toplu model kombinasyonu kararlılık geliştirme ve performans artırmayı destekler. Sınıflandırıcı modelin temel parçalarıyla sınırlıdır.



Şekil 2.7: MOE Model Mimarisi

MoE, sınıflandırma görevini alt görevlere böler ve her alt görev için bir uzman geliştirir. Uzmanlardan gelen bilgilerin birleşimiyle tahmine dayalı modelleme problemi için daha güvenilir sonuçlara ulaşmayı sağlar.

MoE modelleri uzman ve geçit ağları olmak üzere iki temel bileşene sahiptir. Yeşil kareler, işlevi girdi verilerinin sınıfını tahmin etmeyi öğrenmek olan(μ_i parametresine sahip) uzman ağları temsil eder. Gri dikdörtgenle sembolize edilen (φ_{i} parametresine sahip) geçit ağı, uzmanların her birine ağırlıklar atar ve böylece nihai sınıflandırmanın yapılmasını sağlar.

MoE modeli, bölümlenmiş girdi verilerini kendisini oluşturan modelleri beslerken kullanır ve olasılıksal bir sınıflandırıcı topluluğu oluşturur. Girdi verilerinin sınıflandırılması için uzmanın önemine göre yapılan ağırlıklandırma sonrası çıktılar toplanır ve değerlendirilir.

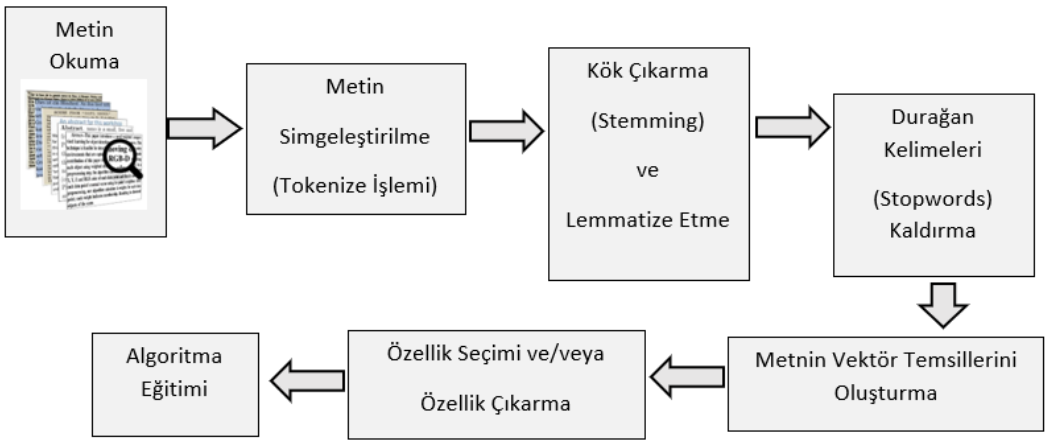
ÜÇÜNCÜ BÖLÜM

METİN SINIFLANDIRMA

Metin Sınıflandırma, bir makine öğrenimi tekniği kullanılarak metin belgesinin önceden tanımlanmış sınıflara ayrıştırılmasını sağlar. Sınıflandırma genellikle metin belgesinden çıkarılan önemli kelimeler veya özellikler üzerinden yapılır.

Metin sınıflandırması, makine öğrenimi teknikleri kullanılarak başarılı bir şekilde otomatikleştirilebilir. Sınıflandırıcının doğruluğunun belirlenmesinde ön işleme ve özellik seçme adımları, sınıflandırıcıya verilen eğitim girdisinin boyutu çok önemli bir rol oynar.

Metin sınıflandırması için genel strateji Şekil 1'de gösterilmektedir. İlgili ana adımlar şunlardır **Şekil 3.1'** de gösterilmiştir.



Şekil 3.1: Metin Sınıflandırma Stratejisi

3.1.Veri Önişleme

Belgeler kelime dizisi olarak değerlendirilirse, her belge genellikle bir dizi sözcükle temsil edilir. Bir belgede sunulan kelimelerin tümü, sınıflandırıcıyı eğitmek için kullanılamaz. Belgeyi temsil eden en uygun boyut ve içeriğe ulaşmak için veri önişleme adımları uygulanmalıdır.

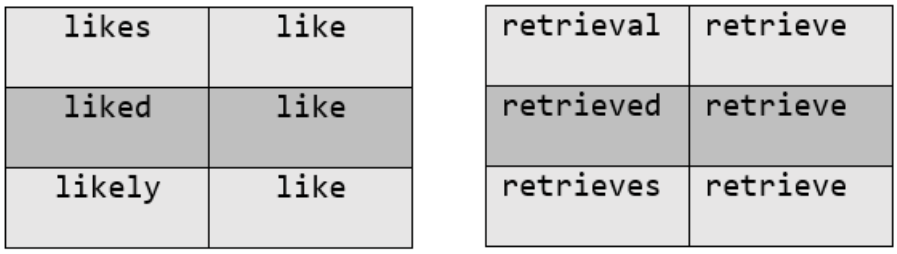
Veri ön işleme, giriş metni belgelerinin boyutunu önemli ölçüde azaltır. Cümle sınırı belirleme, metnin diline özgü durağan sözcükleri çıkarma ve kök çıkarma gibi etkinlikleri içermektedir.

Belgelerin çoğunda yardımcı fiiller, bağlaçlar, edatlar gibi gereksiz kelimeler vardır. Bu kelimelere Durağan Kelimeler (Stopwords) denir.

Verilerdeki gürültüyü ortadan kaldırmak ve verilerin tutarlı bir şekilde temsil edilmesini sağlamak için çeşitli adımlar uygulanır. ‘a’, ‘an’, ‘the’ vb. gibi

İngilizcedeki durağan sözcüklerin kaldırılması, tüm sözcüklerin küçük harfe indirgenmesi, tüm bağlantılar, sayılar ve karakterler gibi ?,:,. vb. kaldırılması gibi işlemler veri önişleme adımlarındandır. Bunlarla birlikte noktalama işaretleri ve fazla boşluklar metni sadeleştirecek şekilde kaldırılır.

Kök çıkarma (Stemming), başka bir yaygın ön işleme adımıdır. İlk özellik setinin boyutunu küçültmek için, yanlış yazılmış veya aynı köke sahip sözcükleri kaldırır. Kök çıkarma, metin analizinde kelime dağarcığını belirlemek için kullanılır.



likes	like
liked	like
likely	like

retrieval	retrieve
retrieved	retrieve
retrieves	retrieve

Şekil 3.2: Porter Stemmer İle Kök Çıkarma Örneği

Lemmatizasyon, kök çıkarmaya benzer, ancak kelimelerin kullanıldığı bağlam ilişkilerini değerlendirir. Böylece benzer anlamlara sahip kelimeleri tek bir kelimeye

bağlar. Kelimelerin cümle içerisinde farklı anlamlarda kullanılma durumları ve sözcükler arasında yapılandırılmış anlamsal ilişkiler bu sayede değerlendirilmiş olur.

Kök çıkarma ve Lemmatizasyon arasındaki fark, lemmatizasyon bağlamı dikkate alarak kelimeyi temel biçimine dönüştürürken, kök çıkarma yalnızca son birkaç karakteri kaldırarak yanlış anlamlara ve yazım hatalarına yol açabilmektedir.

Örneğin Lemmatizasyon, ‘careing’ in temel formunu ‘care’ olarak doğru bir şekilde tanımlarken, kök çıkarma, 'ing' kısmını kesip ‘car’ arabaya dönüştürür.

Bir eğitim kümesindeki tüm sözcüklerin kümesine sözcük dağarcığı veya özellik kümesi denir. Bir belge, bir özellik sözcüğünü içeriyorsa 1, içermiyorsa 0 değerini atayan bir ikili vektör tarafından sunulabilir. Metnin vektör temsilleri bu şekilde oluşturulur.

Özellik çıkarma yöntemlerinin amacı, sınıflandırmada faydalı olmadığı düşünülen özellikleri kaldırmak böylece veri kümesinin boyutunun azaltılmasını sağlamaktır.

Özellik çıkarma genellemeyi arttırmayı sağlar. Genelleme, bir sınıflandırıcının kurucu özellikler yerine eğitim verilerinin koşullu özelliklerine göre kategori ayarlamasının azaltılması olarak tanımlayabiliriz.

Özellik seçimi yaklaşımı, özellik çıkarma yaklaşımlarından farklıdır, fakat benzer olarak amacı özellik setinin boyutunu azaltmaktır. Özellik çıkarma, orijinal özelliklerden yeni özellikler oluştururken, özellik seçimi ise mevcut özelliklerin en iyi temsili olan bir alt küme oluşturur.

Temel Bileşen Analizi (PCA), özellik seçimi için bilinen bir yöntemdir. PCA mümkün olduğu kadar çok bilgiyi koruyan bir özellik seçimi yapar, varyansın çoğuna sahip olan özellikleri seçerek özellik sayısını azaltır.

PCA değişkenlerin en iyi alt kümesiyle veri kümesinin tamamı arasındaki fikir birliğini sağlamak için sırasıyla:

Verilerin kovaryans matrisini oluşturur. Bu matrise ait özvektörleri çıkarır ve özvektörleri, orijinal verinin büyük bir varyans oranını oluşturmak için kullanır.

3.2.Özellik Çıkarma

3.2.1. TF-IDF

TF-IDF (Terim Frekansı- Ters Belge Frekansı) kelimenin ait olduğu belgeyi ne kadar temsil ettiğinin ölçüsüdür. Hem kelimenin belgedeki sıklığı olan TF'yi hem de kelimenin belgelerdeki dağılımı olan IDF'yi birleştirerek hesaplanır.

TF (terim sıklığı) bir kelimenin ne kadar önemli olabileceğinin ölçüsüdür. Metin içerisinde kelimenin kaç kez tekrar ettiğinin sayısı kelimenin ağırlığı ile doğru orantılıdır.

TF belirlenirken sadece sıklığın dikkate alınması, bir belgede birçok kez geçen ancak önemli olmayan sözcüklerin (stopwords) ağırlıklarının fazla olmasını gerektirir. Bu durumun önüne IDF ile geçilir.

IDF kelimeyi içeren belgelerin logaritmik olarak ölçeklendirilmiş ters kesri olarak tanımlanır ve yaygın sözcüklerin, nadiren ortaya çıkan sözcüklere göre daha değersiz olabileceği fikrine sahiptir.

Bir terimin metinde görülme sayısı arttıkça, TF oranı 1'e yaklaşır ve IDF ve TF-IDF'yi 0'a yaklaştırır. Böylece yaygın sözcüklerin ağırlığını azaltırken, sık kullanılmayan sözcüklerin ağırlığını da artırır.

Bir kelime için TF-IDF değeri, TF ve IDF değerlerinin çarpılmasıyla hesaplanır.

$$\mathrm{TF}(\mathrm{T},\mathrm{D}) = \frac{\text{T Teriminin D Belgesindeki Sıklığı}}{\text{D Belgesindeki Toplam Terim Sayısı}} \quad \textbf{(3.1)}$$

$$\mathrm{IDF} = \log \frac{\text{Toplam Belge Sayısı}}{\text{T Teriminin Tüm Belgelerdeki Sıklığı}} \quad \textbf{(3.2)}$$

$$\mathrm{TF-IDF}(\mathrm{T},\mathrm{D}) = \mathrm{TF}(\mathrm{T},\mathrm{D}) \times \mathrm{IDF}(\mathrm{T}) \quad \textbf{(3.3)}$$

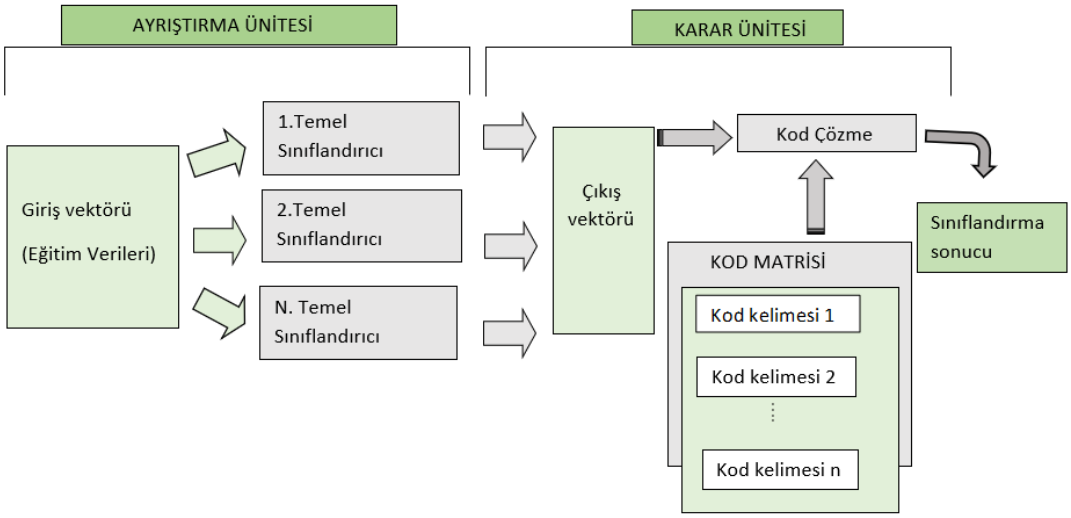
DÖRDÜNCÜ BÖLÜM

ARAŞTIRMA VE YÖNTEMLER

4.1.Hata Düzelten Çıkış Kodları (Error Correcting Output Codes)

ECOC, çok sınıflı sınıflandırma problemi için tasarlanmış bir kolektif öğrenme yöntemidir. Temelde, çok sınıflı bir sorunu çözmek için birkaç ikili sınıflandırıcıyı birleştirerek çalışır.

ECOC sınıflandırıcısında, belirli bir veri kümesinin her sınıfına bir kod sözcüğü atanır. İletilen veya depolanan bilgiler kod sözcükleri tarafından kodlanır. Kod sözcükleri bit hatalarının kolaylıkla çözülmesini sağlayan Hamming mesafesine sahiptir.



Şekil 4.1: ECOC Model Mimarisi

ECOC, çok sınıflı bir problemi ikili alt problemlere ayrıştırma avantajına sahiptir. Her bir alt problem, bir ikili sınıflandırıcı tarafından ele alınır ve çok sınıflı problem için nihai çözüm, ikili sınıflandırıcılardan gelen sonuçların toplanmasıyla oluşturulur.

Bu sayede ECOC, diğer sınıflandırıcıların normalde zorlandığı çok sınıflı problemlerde daha iyi performans gösterebilir.

ECOC çerçevesi kodlama ve kod çözme olmak üzere iki aşamadan oluşur.

Kodlamada, kodlama matrisindeki her sütunun bir ikili sınıflandırıcıyı temsil eder. Kodun uzunluğu, koddaki sütunların sayısıdır. Koddaki satır sayısı, çok sınıflı

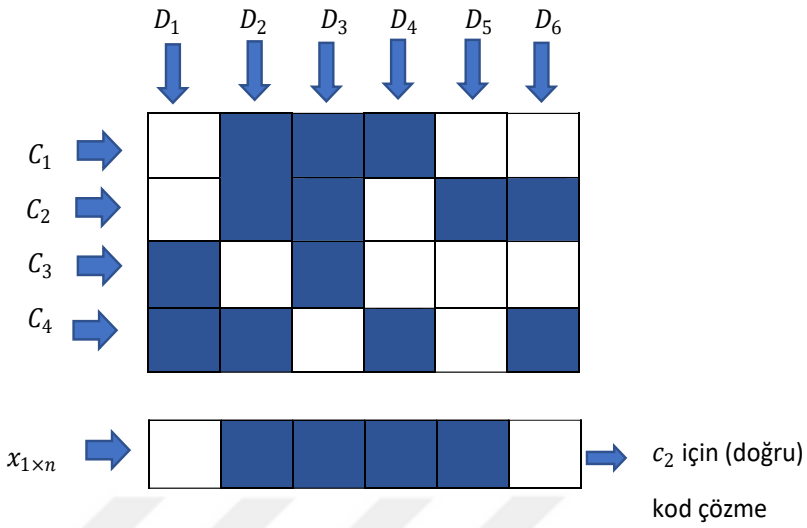
öğrenme problemindeki sınıf sayısına eşittir. Bu matrisin satırlarına kod sözcükleri adı verilir ve her kod sözcüğü bir sınıfı belirtir. İkili sınıflandırıcı eğitim yöntemleri kullanılarak bir temel sınıflandırıcıya, belirli bir sınıfı diğerlerinden ayırma görevi verilir. Bu görev sınıfların rastgele bir ikiliğini öğrenme görevi olarak açıklanabilir.

Her sınıf için bir kod sözcüğü oluşturur. Kod sözcüklerini matrisin sıraları olarak düzenleyerek, "kodlama matrisi" M tanımlanır. M matrisi, her sütun için bir tane olmak üzere n adet (kodun uzunluğu kadar) ikili öğrenme problemi kümesi olarak yorumlanır. Her sütun, sınıfların bir bölümünü tanımlar. Eğer sınıf üyeliği varsa +1,yoksa -1 ile kodlanır.

ECOC' un kodlama adımından sonra elde edilen M kodlama matrisi, her bir sınıflandırıcı çıkışının {+1, -1} olduğu ikili olabilir ve tüm girdi verisi sınıflarını iki sınıfa sınıflandırır. n adet ikili sınıflandırıcının çıktıları sonucunda test setindeki her veri noktası için bir kod elde edilir. Bu kod, M matrisinde tanımlanan her sınıfın temel kod sözcükleri ile karşılaştırılır ve veri noktası “en yakın” kod sözcüğüne sahip sınıfa atanır.

Kod çözme aşamasında, çıkış vektörü karşılaştırılırken en yakın kod sözcüğünü bulmak için çıktı vektörünü kodlama matrisi kod sözcükleri ile karşılaştırılır. Girdinin en yakın anahtar kelimeye yakınlığının nasıl tanımlanacağını seçmenin birkaç yöntemi vardır.En yakın kod sözcüğünü seçmek, sistemin temel sınıflandırıcılarına ait bazı hatalarını da düzeltmesini sağlar. Kod çözme aşamasında, verilen bir girdi için temel sınıflandırıcıların çıktıları elde edilir ve girdi, en yakın kod sözcüğüne sahip sınıfa atanır (Ko & Kim, 2005).

Şekilde çıkış vektörü c2 sınıfına sınıflandırılır ve böylece dördüncü temel sınıflandırıcı D4'ün hatasını düzeltir. Ayrıca, en yakın kod kelimesini bulmak için Hamming mesafesi, Öklid mesafesi vb. gibi çok sayıda strateji vardır.



Şekil 4.2: ECOC Örneği

Kod çözme (sınıflandırma), sınıflandırıcı tarafından tahmin edilen kod sözcüğü ile Hamming mesafesine en yakın sınıf kod sözcüğünü eşleştirerek gerçekleştirilir. Temelde ECOC, bire bir ve bire karşı tüm sınıflandırma tekniklerinin genelleştirilmesidir ve bir topluluk tekniği olarak, ikili sınıflandırıcılar rastgele bir örnek üzerinde bağımsız hatalar yaptığında en etkilidir.

Kod çözme sürecinde hata düzeltmeye yardımcı olması için, kod matrisi, farklı sınıflardaki kod sözcükleri arasında büyük bir Hamming mesafesine sahip olacak şekilde tasarlanmalıdır. Her sınıf, ikili uzunluk vektörü olarak kodlanır.

$$\mathrm{HD}(x, y_i) = \sum_{j=1}^{n}(1 - \mathrm{sign}(x^j * y^j))/2 \qquad \textbf{(4.1)}$$

N:(N negatif olmayan bir tamsayıdır) sınıflandırıcıların çıktı etiketleri, a_{l*N} : N uzunluğunda ikili vektörlerden oluşan M matrisi olmak üzere C_0^{*j}, C_1^{*j} şu şekilde tanımlanmıştır:

$$C_0^{*j} = \bigcup_{\substack{i=1^l \\ a_{ij}=0}} C_i^* \quad , \quad C_1^{*j} = \bigcup_{i=1} a_{ij} = 1^l C_i^*$$

$j = \{1, \dots, n\}$ her sütunda, bir ikili algoritma C_0^{*j} ve C_1^{*j} öğrenme kümesi olarak alır ve bazı hipotezler döndürür.

$h_j: S \rightarrow \{0,1\}, x \in S,\ \beta_j = h_j(x), \beta_{\hat{x}} = (\beta_1, \beta_2, \dots, \beta_n)$ ikili vektörü için $\hat{x}$ nesnesi, t'nin aşağıdaki şekilde tanımlandığı t sınıfına sınıflandırılacaktır.

$$t = \arg \min_{i=1,\dots,l} \sum_{j=1}^{N} \left| a_{ij} - \beta_j \right| \qquad \textbf{(4.2)}$$

$\hat{x}$ nesnesi, hamming mesafesi vasıtasıyla matrisin en yakın satırına karşılık gelen bir sınıfın elemanı olarak sınıflandırılır dikkate alınan yaklaşımın sınıflandırma doğruluğu matris seçimine bağlıdır (Danoyan, 2017).

4.1.1. Kodlama yöntemleri

Çok sınıflı sınıflandırma problemini çoklu ikili sınıflandırma problemleri olarak yeniden çerçevelendirirken kullanılabilecek iki yaygın yöntem, bire karşı kalan (OVA) ve bire karşı bir (OVO) tekniklerini içerir.

4.1.1.1.Bire Karşı Bir (One vs One)

Bire karşı bir kodlama yönteminde, öncelikle çoklu sınıf problemini çoklu ikili sınıf problemine ayrıştırılır. Bu ikili sınıflar, tüm sınıfların birbiriyle eşleştirilmesiyle oluşturulur. Sınıflandırıcılar her sınıf çiftini birbirinden ayırması için eğitilir. k sayıda sınıf için k(k-1)/2 ikili sınıflandırıcı kullanılır. Her sınıflandırıcı, her bir çiftin eğitim verileriyle eğitilir.

Bire bir kodlama matrisi, birkaç ikili sınıflandırıcının birleşimidir. Her sınıflandırıcıda üç öğe (-1, 0, +1) vardır, bir sınıf pozitif, diğerleri negatif sınıflardır. “0” değeri ise yok sayılacağı anlamına gelir.

Tablo 4.1: 4sınıflı 6 sınıflandırıcılı OVO Örneği

	h_1	h_2	h_3	h_4	h_5	h_6
c_1	+1	+1	0	0	0	+1
c_2	-1	0	0	-1	+1	0
c_3	0	+1	0	-1	0	-1
c_4	0	0	0	+1	+1	0

4.1.1.2.Bire Karşı Kalan (One vs All)

Bu strateji, sınıf başına bir ikili sınıflandırıcı yerleştirmeyi içerir. One-vs-All kodlama yönteminde, çoklu sınıf problemini yine çoklu ikili sınıf problemine ayrıştırılır. Sınıflandırıcıları, bir sınıfı diğer sınıflardan ayırmak için eğitilir.

Bu yaklaşımda k sayıda sınıflandırıcı için k sayıda sütun içeren bir ECOC matrisi vardır. Her sınıflandırıcı, bir sınıf pozitif olarak ve diğer tüm sınıflar negatif girdiler olarak eğitilir. Böylece her sınıflandırıcı bir sınıfı diğer tüm sınıflardan ayırır. ECOC'de tüm köşegen elemanlar +1 ve geri kalanlar -1'dir.

Tablo 4.2: **5 sınıflı 5 sınıflandırıcılı OVA kod matrisi örneği**

	h_1	h_2	h_3	h_4	h_5
c_1	+1	-1	-1	-1	-1
c_2	-1	+1	-1	-1	-1
c_3	-1	-1	+1	-1	-1
c_4	-1	-1	-1	+1	-1
c_5	-1	-1	-1	-1	+1

4.2.İlgili Çalışmalar

Hata düzelten çıktı kodlarının öncülüğü Dietterich ve arkadaşları tarafından yapıldı (1995). Öğrenme algoritmasını farklı kullanarak her bir biti ayrı ayrı ECOC matrisinde kodladılar ve sınıflara ait kodlarla karşılaştırarak en uygun sınıfa üyelik verdiler. Elde ettikleri sonuçlarla ECOC yaklaşımının güvenilir sınıf olasılık tahmini üretebileceğini ve sistem hatalarının azaldığını gösterdiler. Küçük örnekli verilerde ECOC

kullanımının etkinliğiyle ilgili değerlendirme de sundular. ECOC kullanılarak öğrenilen karar ağaçlarının büyüklüğü ve karmaşıklığı göz önüne alındığında, karmaşık ağaçların güvenilir bir şekilde öğrenilmesi için genellikle daha fazla veri gerektirdiğinden, küçük örneklem boyutlarında ECOC yönteminin de iyi performans göstermeyebileceğini belirttiler. Ayrıca hata düzeltme kodundaki kod sözcüklerinin, sınıflara keyfi olarak atanmasının etkilerini de araştıran Dietterich ve arkadaşları, farklı rasgele atamaların performansta istatistiksel olarak anlamlı bir değişikliğe sebep olmadığını göstermişlerdir (Dietterich & Bakiri, 1995).

Dietterich ve Bakiri, başka bir çalışmada çok sınıflı öğrenme problemlerine yönelik üç yaklaşımı ECOC'la karşılaştırdı. ECOC'un, çok sınıflı problemlerde performans iyileştirmede diğer yöntemlere kıyasla daha etkili olduğunu belirtmişlerdir. Bununla birlikte ECOC'un karar ağaçlarını ve sinir ağlarını iyileştirdiğini deneyimlediler. Ek maliyetlere sebep olmasından dolayı ECOC önerilmeyen durumları da belirttiler. ECOC kullanılarak öğrenilen karar ağaçlarının genellikle daha büyük ve daha karmaşık olması, ECOC kullanılarak öğrenilen sinir ağlarının daha uzun ve daha dikkatli eğitim gerektirmesi, eğitimin hızlı ve tamamen özerk olması gereken alanlarda eğitim sırasında zorluklarla karşılaşma potansiyeli nedeniyle bahsedilen durumlar için ECOC kullanımını önermemişlerdir (Dietterich & Bakiri, 1995).

Kong ve Dietrich, karar ağacı öğrenme algoritmaları ile kullanıldığında ECOC tekniğinin etkinliğini değerlendirmiştir. ECOC'un öğrenme algoritmasının varyansı azaltabileceğini ve algoritmanın yanlılığından kaynaklanan hataları düzeltebileceğini göstermişlerdir (Kong & Dietterich, 1995).

Çok sınıflı öğrenme problemlerinde kullanılan topluluk tabanlı sistemlerin genel sınıflandırma performansı üzerindeki etkilerine odaklanan Tumer ve arkadaşları özdeş olmayan eğitim setleri üzerinde çalıştı. Yaptıkları deneylerle topluluk sınıflandırma oranının artması için birleştirilen sınıflandırıcılar arasındaki korelasyonun azaltılmasının sonuçlar üzerindeki olumlu etkisini gösterdiler. Bununla birlikte bireysel sınıflandırıcıların eğitim verilerinin alt kümelerini kullanmaları, eğitim seti boyutu küçüldükçe sunulan alt kümede eksiliğe sebep olabileceği için yeterince büyük olmayan veri setleri için bireysel sınıflandırıcı performansının bozulabileceğini belirtmişlerdir (Tumer & Ghosh, 1996).

Schapire, arttırma ve ECOC'u birleştirmeye dayalı hibrit bir yöntem önerdi. Zayıf öğrenicilerin güçlerini birleştirerek sınıflandırma algoritmasının doğruluğunu iyileştiren arttırma yöntemi ile ECOC u birleştirerek çok sınıflı öğrenme problemlerinin çözümünde kullandı. Birleştirilmiş bu hipoteziyle eğitim ve genelleme hatası konusunda daha etkili sonuçlar elde etti. Yeni algoritmadan elde edilen sonuçları diğer oylama yöntemleriyle de kıyasladı ve etkinliğini kanıtladı. Önerdiği yeni yöntem daha az programlama çabası gerektirebildiği gibi daha hızlı da çalışabilmektedir (Schapire, 1997).

Benzer olarak Zhang ve arkadaşları, diskriminant özniteliklerin seçilmesi ve güçlü iki sınıflı sınıflandırıcı oluşturmak için zayıf sınıflandırıcıları kullanan Adaboost'u, çoklu sınıf problemini iki sınıflı sınıflandırma problemlerinden oluşan bir gruba indirgemek için ECOC tabanlı yöntemi birlikte kullanmıştır. Altı sınıflı bir nesne sınıflandırma problemi üzerinde elde ettikleri deneysel sonuçlarla, bu yaklaşımın farklı verilerde gerçek zamanlı ve sağlam nesne sınıflandırması sağlayabildiğini göstermişlerdir (L. Zhang vd., 2007)

Eskalera ve arkadaşları, Alt sınıf ECOC stratejisini önerdiler. Yöntem orijinal sınıf setini alt sınıflara bölmek ve sonrasında ikili problemleri ECOC tasarımı yardımıyla çözmeye dayanmaktadır. Temel sınıflandırıcının sınıfları ayırt edecek kadar esnek olmadığı durumlarda, bir kümeleme yaklaşımı olarak alt sınıf stratejisi etkili olmuştur. Veri seti üzerinde farklı temel sınıflandırıcılardan elde edilen sonuçlarla ECOC tasarımının karşılaştırıldığında, Eskalera ve arkadaşları çoğu durumda alt sınıf stratejisinin önemli performans iyileştirmeleri elde edebildiğini göstermişlerdir (Escalera vd., 2008).

Zor ve arkadaşları, yanlılık ve varyans çerçevesi kullanarak önyükleme toplama (Torbalama) ve Hata Düzeltme Çıktı Kodlama (ECOC) topluluklarını analiz ettiler. Tekli sınıflandırıcılarla kıyaslandığında Torbalama ve ECOC' la elde edilen tahmin hatalarının düşük olduğunu bunun varyans ve yanlılıktaki azalma sonucunda olduğunu belirttiler. Bu gözlem sayesinde yanlılığın ve varyansın tahmin hatasına katkılarının topluluklar kullanıldığında daha küçük olduğu sonucuna varmışlardır (Zor vd., 2011).

Bagheri ve arkadaşları, ECOC la ilişkili yeni bir yaklaşım önerdi. ECOC matrisinin tasarlanmasındaki önemli bir faktör olan ikili sınıflandırıcılar arasındaki bağımsızlığı geliştirmek için üç boyutlu kod matrisi tasarladılar. Kaba küme tabanlı öznitelik

seçimini kullanarak Kaba Küme Altuzay ECOC (RSS-ECOC) isimli algoritmayı önerdiler. RSS-ECOC'la topluluk öğrenmesini oluşturan temel öğrenenlerin çeşitliliğini etkili bir şekilde kullanmayı amaçladılar. Topluluk öğrenmesinde her öğreniciyi farklı özellik alt kümelerinden oluşan verilerle eğitmek, varyansı azaltmayı sağlar. Bagheri ve diğerlerinin RSS-ECOC da oluşturdukları matris kodu, bağımsız sınıflandırıcılar oluşturmak için farklı rasgele alt uzaylar kullanılarak tasarlandı. Bağımsız sınıflandırıcılar sayesinde de önemli performans iyileştirmeleri elde etmişlerdir (Bagheri vd., 2012).

Suzuki ve arkadaşları, ikili sınıflandırıcıları, sınıflandırma için kullanılmayan kategorilerle birleştiren yeni bir yapılandırma yöntemi olarak RM (Reed-Muller) kodu kullanan ECOC yaklaşımını önerdi. RM kodunda, her kategoriyi temsil eden kod sözcükleri arasındaki Hamming mesafesinin büyük olması ve her sınıflandırıcı arasında eşit olması sayesinde sınıflandırıcıların yüksek bir sınıflandırma doğruluğu ile birleştirilebildiğini göstermişlerdir. Önerdikleri yöntemin, ikili sınıflandırıcılar için iki kategori seti arasındaki eğitim verilerinin dengesini iyileştirmiştir. Üçlü kod tablosu kullanarak her bir sınıflandırıcı için daha az eğitim verisine ihtiyaç duyması, eğitim verisi miktarını azaltmayı da sağlamıştır. Bu sayede hesaplama karmaşıklığının azaltılabileceğini göstermişlerdir (Suzuki vd., 2017).

Kumoi ve arkadaşları, ECOC'un teorik performansını değerlendirmek ve ECOC' u oluşturan ikili sınıflandırıcıların en iyi kombinasyonunu netleştirmek için bir çerçeve önerdiler. Yaptıkları çalışmada, hata teriminin normal dağılımda olduğunu varsayarak kod kelime tablosunun performansı analiz ettiler. Elde ettikleri sonuçlarla, kod sözcükleri arasındaki Hamming mesafesinin artırılmasının hata oranını azaltabileceği teorik olarak açıklığa kavuşturulmuştur (Kumoi vd., 2022).

BEŞİNCİ BÖLÜM

DENEYLER

5.1.Veri Setleri

5.1.1. Nefret Söylemi ve Saldırgan Dil İçeren Tweetlerden Oluşan Veri Seti

Bu çalışmada kullanılan ilk veri seti Kaggle'dan alınmıştır. İngilizce tweetlerden oluşan veri seti Crowdflower kullanıcıları tarafından etiketlenmiştir. 24.783 tweet özelliklerine göre üç ana kategoriye ayrılır: Nefret söylemi, Saldırgan dil ve Normal. Geniş bir tanıma sahip olması ve nefret söylemi ile karıştırılabilmesi nedeniyle saldırgan dile ait örnek sayısı diğer iki kategorideki örnek sayısından daha fazladır. Üç sınıfa ait tweetlerin metin uzunluğu 0-200 karakter arasındadır. Tweetlerin dağılımı şu şekildedir: %5,7 nefret söylemi, %77,4 saldırgan dil ve %16,7 normal sınıf (Mercan vd., 2021).

Makine öğrenimi algoritmalarıyla işlenmesini mümkün kılmak için her sınıfa sayısal bir etiket atandı. Veri seti %80 eğitim ve %20 test alt kümelerine ayrılmıştır.

Durağan kelimeleri çıkarmak için Stopwords kullanıldı. Veri seti tweetlerden oluştuğu için '#rt', 'ff' gibi tweetlere ait olan ekleri ve emojileri çıkaracak şekilde Stopwords genişletildi.

Porter Stemmer, farklı zamanlara göre çekimlenerek kullanılmış fiilleri kök olarak ele almayı ve türetilmiş kelimelerin de köklerine indirgenmesini sağlamak için kullanıldı.

İngilizce dili için geniş bir sözcük veritabanı olan WordNetLemmmatizer, sözcükler arasında yapılandırılmış anlamsal ilişkiler kurmak için kullanıldı ve veri seti lemmatize edildi.

Metin verisinin sayısal gösterime (özellik çıkarma) dönüştürülmesi için TF-IDF özellik çıkarma yöntemi kullanıldı.

Öğrenme algoritmalarının performansı, verilerin temsil şeklinden etkilenir. Özellik çıkarımı için çeşitli algoritmalar önerilmiştir. TF-IDF etkili sonuçlar göstermiştir; bu nedenle, bu çalışmada özellik çıkarımı için kullanıldı. Sıkıştırılmış seyrek satır biçiminde 245461 depolanmış öğeli, 24783x8144' lik TF-IDF matrisi elde edildi.

5.1.2. Bilgisayar Bilimleri-Matematik Veri Seti

Veri seti, Matematik ve Bilgisayar Bilimleriyle ilgili Youtube videolarından alıntılanan altyazıların derlenmesiyle oluşturulmuştur.

860 adet dersin alt başlıklarından oluşturulan veri seti, Lineer Cebir, Hesaplama (Calculus), Olasılık, Bilgisayar Bilimleri, Algoritmalar, Diferansiyel Denklemler, Yapay Zeka, Mühendislik için Matematik, Veri Yapıları, Statik, Doğal Dil İşleme isimli 11 adet sınıfa sahiptir. En yüksek sayıda veriye sahip olan sınıflar sırasıyla Lineer Cebir ve Olasılık iken en az sayıda veri Doğal Dil İşleme sınıfına aittir.

Noktalama ve büyük harf kullanımının kaldırılması, belirteçleştirme (tokenizasyon), durağan kelimelerin kaldırılması, Porter Stemmer'la kök çıkarma (stemming) gibi veri önişleme adımları uygulandı.

Benzer şekilde metin verisinin sayısal gösterime TF-IDF özellik çıkarma yöntemi ile dönüştürüldü. Sıkıştırılmış seyrek satır biçiminde 821615 depolanmış öğeli 860x10000' lik TF-IDF matrisi elde edildi.

5.2.Kullanılan Modeller

5.2.1. Yerel Modellerin Uygulanması

Scikit-Learn Python kitaplığı kullanılarak Geleneksel Makine Öğrenimi modellerinden SVM, LR, RF ve NB Saldırgan Dil ve Nefret Söylemi veriseti üzerinde uygulandı.

SVM için, daha düşük boyutlu girdinin daha yüksek boyutlu uzaya dönüştürülmesi için çekirdek olarak radyal tabanlı fonksiyon kullanıldı.

Gamma (γ) 0,05 olarak ayarlanırken ceza parametresi için optimal değeri 150 olarak elde ettik. Sınıflandırma için, basitlik ve yüksek doğruluk nedeniyle OVA (bire karşı hepsi) modeli kullanıldı.

LR modeli için l_2 ile çok terimli lojistik regresyon düzenlileştirme uygulandı. LR'nin ürettiği sonuçlar, temel olarak h(Z) olasılığı olan [0, 1] arasındaydı. Her girdi örneği için nihai çıktıyı elde etmek için çıktıyı bir lojistik fonksiyondan geçirdik.

RF sınıflandırıcının iki önemli parametresini; tahminci sayısı 50 olarak ve ağacın maksimum derinliği 25 olarak ayarladık.

5.2.2. Hata Düzelten Çıkış Kodlarının Uygulanması

Ecoc uygulaması için Scikit-Learn kitaplığı çok sınıflı sınıflandırma stratejilerinden Hata düzeltme çıkış kodu (Sklearn.multiclass. OutputCodeClassifier) kullanıldı. Bu strateji seçilen ikili sınıflandırıcı ile çok sınıflı bir sınıflandırma probleminin öğrenilmesini sağlar. Her sınıf için 0'lar ve 1'lerden oluşan bir kod oluşturur ve bir kod kitabında saklar. Her sınıflandırıcıyı, kod üzerinde eğitir.

Sınıflandırılmak üzere bir örnek geldiğinde etiketin kodunu kod kitaplığından geri alır. Kod kitabındaki kodlarla karşılaştırır "en yakın" olan etiketi sınıf olarak seçer. Yakınlığı Hamming mesafesi cinsinden belirler.

Kullanılan veri setlerine ECOC uygulandı. Çok sınıflı sınıflandırma problemini alt ikili sınıflandırma problemlerine indirgeyerek çözmek üzere ikili sınıflandırıcı olarak Lojistik Regresyonu kullandık. Bağımsız değişken ve yanıt değişkeni arasındaki ilişkiyi belirlemede etkili sonuçlar vermesi ve uygulama kolaylığı sebebiyle Lojistik Regresyon tercih edilmiştir.

Stratejiye ait parametreleri ayarlarken, her sınıf için kodlanacak bit sayısının uzunluğunu belirlemek üzere code_size özniteliğini 1den 10'a kadar tamsayılar olarak seçtik. Bu sayede en yüksek doğruluk değeri için en ideal code_size belirlendi.

Model daha sonra üç tekrarlı (n_repeats=3) ve 10 katlı (n_splits=10) tekrarlanan katmanlı k-katlı çapraz doğrulama kullanılarak değerlendirildi. Tüm tekrarlar ve kıvrımlar boyunca sınıflandırma doğruluğunun ortalaması ve standart sapma kullanılarak modelin performansı incelendi.

ALTINCI BÖLÜM

DEĞERLENDİRME VE SONUÇ

Modeller Doğruluk (Ac), Kesinlik (Pr), Duyarlılık (Rc) ve F1-Measure (F1) açısından değerlendirildi. Bu standart metrikler şu şekilde hesaplanır:

TP (True Positive – Doğru Pozitif): Doğruyu doğru olarak değerlendirmek.

TN (True Negative – Doğru Negatif): Yanlışı yanlış olarak değerlendirmek.

FP (False Positive – Yanlış Pozitif): Yanlışı doğru olarak değerlendirmek.

FN (False Negative – Yanlış Negatif): Doğruyu yanlış olarak değerlendirmek.

Doğruluk, modelin doğru tahminlerinin ölçüsüdür.

$$\text{Doğruluk(Ac)} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad \textbf{(6.1)}$$

Kesinlik doğru sınıflandırılmış pozitif örneklerin sayısının tahmin edilen tüm örneklerin sayısına oranıdır. Kesinliğin artması doğru olarak değerlendirilen yanlışların azalmasıyla mümkün olur. Bu durum pozitif olarak etiketlenmiş bir örneğin gerçekten pozitif olmasını gerektirir.

$$\text{Kesinlik(Pr)} = \frac{\text{TP}}{\text{TP+FP}} \quad \textbf{(6.2)}$$

Duyarlılık, sınıfın ne kadar doğru tanındığını gösterir. Toplam pozitif değerlendirmeler içerisindeki gerçek pozitif değerlendirmelerin oranı arttıkça duyarlılık artar.

$$\text{Duyarlılık(Rc)} = \frac{\text{TP}}{\text{TP+FN}} \quad \textbf{(6.3)}$$

F1 ölçütü, kesinlik ve duyarlılığın ağırlıklı ortalaması alınarak hesaplanır. Eşit olmayan sınıf dağılımları söz konusu olduğunda doğruluktan daha kullanışlı olabilir. Yanlış pozitiflerin ve yanlış negatiflerin değerinin çok farklı olduğu durumlarda kesinlik ve duyarlılığın birlikte değerlendirilmesi daha faydalı olabilir.

$$F_1 = 2 \times \frac{\text{Kesinlik} \times \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad \textbf{(6.4)}$$

Geleneksel makine öğrenimi modelleri için elde edilen genel sonuçlar **Tablo 6.1**'de özetlenmiştir. Nefret söylemi ile saldırgan dilin birbirine yakın olması ve çoğunlukla birbirinin yerine kullanılması, saldırgan dilin nefret söylemine benzer bazı cümleleri içermesi nefret söylemi için sınıflandırma doğruluğunun düşük olmasını açıklayabilir. En yüksek doğruluk, RF (%90,31) sınıflandırıcı tarafından elde edildi, bunu LR (%89,75), SVM (%89,32), ve NB (%64,91) izledi.

Tablo 6.1: **Yerel Modellerin Doğruluk Tablosu**

Model	**Doğruluk (Accuracy)**
Lojistik Regresyon (LR)	89.75
Random Forest (RF)	90.31
Saf Bayes (NB)	64.91
Destek Vektör Makinesi (SVM)	89.32

Tablo 6.2: ECOC Code Size Tablosu

Code_size	Doğruluk(Accuracy)	Doğruluğa Etkisi
i=1	77.8	0.002
i=2	88.6	0.005
i=3	89.5	0.004
i=4	89.4	0.005
i=5	89.5	0.005
i=6	89.5	0.004
i=7	89.5	0.005
i=8	89.5	0.004
i=9	89.5	0.004

ECOC'un çok sınıflı veri setleriyle daha etkin çalışmasına karşın kullanılan veri setinin yeterli sayıda sınıfa sahip olmaması ve verisetini oluşturan sınıflardaki veri sayısının dengesizliği (%5,7 nefret söylemi, %77,4 saldırgan dil ve diğer) ECOC'un etkinliğini azaltmış olabilir.

ECOC un ayrık sınıflara sahip veri setlerinde etkili sonuçlar vermesine karşın kullanılan veri setini oluşturan sınıfların (nefret söylemi ve saldırgan dil) birbirlerine benzer içeriklerden oluşması ve çoğu zaman birbirinden ayırmanın zor olması ECOC kullanımının performans üzerindeki olumlu etkisini azaltmış olabilir.

Bilgisayar Bilimleri-Matematik veri setinin kullanılan diğer veri setine göre çok sınıflı olması daha etkili sonuçlar almamızı sağladı. Benzer şekilde sınıfların birbirinden net olarak ayrılmayışı sınıflara ait kavramların ve başlıkların ortak terimler içermesi sınıf ayrımının çok bariz olmaması ECOC un performansına olumsuz etki etmiştir.

Çok sınıflı, sınıflara ait örnek sayısının benzer olduğu ve sınıf ayrımının net olduğu veri setleri üzerinde ECOC etkili performans gösterebilir.

Bilgisayar Bilimleri-Matematik veri setinin Lojistik Regresyon da ECOC uygulaması sonuçları aşağıdaki gibidir.

Tablo 6.3: ECOC Code Size Tablosu

Code_size	Doğruluk(Accuracy)	Doğruluğa Etkisi
i=1	11.4	0.025
i=2	14.8	0.036
i=3	13.2	0.028
i=4	13.7	0.035
i=5	12.7	0.026
i=6	15.0	0.028
i=7	13.9	0.029
i=8	15.0	0.031
i=9	14.4	0.032

6.1.Öneriler

Çalışmamızda sınıf sayısının üçten fazla olduğu çok sınıflı bir sınıflandırma problemi için ECOC modelinin Lojistik Regresyon la üç sınıflı diğer veri setine göre daha etkili sonuçlar elde ettiğini gösterdik. Gelecekteki bir çalışma, daha büyük hacimde veriye sahip sınıflandırma problemlerinin diğer sınıflandırma algoritmalarıyla veya derin öğrenme metoduyla çözülmesinde ECOC modelinin uygulamalarına dair olabilir.

Çalışmada kullanılan veri setindeki sınıflara ait veri sayısının dengesizliği ECOC' un etkinliğini azaltmıştır. İlerideki bir çalışma sınıf dengesizliği olmayan veri setleri için daha iyi sonuçlar almak üzere olabilir.

Gelecekteki bir başka önemli çalışma, ECOC kullanımı için tasarlanmış kütüphanenin güncellenmesiyle daha etkili sonuçlar elde etmeyi sağlayabilir. Mevcut haliyle az sayıda parametreye sahip kütüphanede değiştirilebilir ve ayarlanabilir tek parametre code_size parametresidir. Kod çeşidi olarak yalnızca rastgele (random) kodların kullanımını destekleyen kütüphane, kod parametresinin değiştirilebilir olmasıyla daha belirgin yapısal özelliklere sahip olan cebirsel kodlar gibi farklı kod tiplerinin kullanılabilmesi seçeneğini sunacak hale getirilebilir. Daha fazla parametreye sahip farklı yapıdaki kodların kullanımının sağlanması daha etkili sonuçlara ulaşmayı sağlayabilir.

KAYNAKÇA

Bagheri, M. A., Gao, Q., & Escalera, S. (2012). Rough set subspace error-correcting output codes. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 822–827. https://doi.org/10.1109/ICDM.2012.124

Breiman, L. (1996). Bagging predictors. *Kluwer Academic Publishers.*, *8*(3), 1–26. https://doi.org/10.3390/risks8030083

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *19*(6). https://doi.org/10.1145/2939672.2939785

Cheng, J., Li, G., & Chen, X. (2019). Research on travel time prediction model of freeway based on gradient boosting decision tree. *IEEE Access*, *7*, 7466–7480. https://doi.org/10.1109/ACCESS.2018.2886549

Collell, G., Prelec, D., & Patil, K. R. (2018). A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*, *275*, 330–340. https://doi.org/10.1016/j.neucom.2017.08.035

Danoyan, H. (2017). On computational complexity of multiclass classification approach ECOC. *International Scientific and Technical Conference on Computer Sciences and Information Technologies*, *2018-March*, 97–100. https://doi.org/10.1109/CSITechnol.2017.8312149

Dhaliwal, S. S., Nahid, A. Al, & Abbas, R. (2018). Effective intrusion detection system using XGBoost. *Information (Switzerland)*, *9*(7). https://doi.org/10.3390/info9070149

Dietterich, T. G., & Bakiri, G. (1995). Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research*, *2*, 263–286. https://doi.org/10.1613/jair.105

Escalera, S., Tax, D. M. J., Pujol, O., Radeva, P., & Duin, R. P. W. (2008). Subclass problem-dependent design for error-correcting output codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(6), 1041–1054. https://doi.org/10.1109/TPAMI.2008.38

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine*

Learning, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Hoi, S. C. H., Jin, R., & Lyu, M. R. (2006). Large-scale text categorization by batch mode active learning. *Proceedings of the 15th International Conference on World Wide Web*, 633–642. https://doi.org/10.1145/1135777.1135870

Jakkula, V. (Washington S. U. (2016). Tutorial on Support Vector Machine. *Special Issue "Some Novel Algorithms for Global Optimization and Relevant Subjects", Applied and Computational Mathematics (ACM)*, *6*(4–1), 1–15. https://doi.org/10.11648/j.acm.s.2017060401.11

Jiang, S., Pang, G., Wu, M., & Kuang, L. (2012). An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, *39*(1), 1503–1509. https://doi.org/10.1016/j.eswa.2011.08.040

Ko, J., & Kim, E. (2005). On ECOC as binary ensemble classifiers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3587 LNAI*, 1–10. https://doi.org/10.1007/11510888_1

Kong, E. B., & Dietterich, T. G. (1995). Error-Correcting Output Coding Corrects Bias and Variance. *Machine Learning Proceedings 1995*, 313–321. https://doi.org/10.1016/B978-1-55860-377-6.50046-3

Kotsiantis, S., & Kanellopoulos, D. (2012). Combining bagging, boosting and random subspace ensembles for regression problems. *International Journal of Innovative Computing, Information and Control*, *8*(6), 3953–3961.

Koulis, T. (2003). *Random Forests : Presentation Summary*. 1–11.

Kovács, L., & Terstyanszky, G. (y.y.). *Diagnosising faults by supervised and unsupervised learning*. *683*, 7–11.

Kumoi, G., Yagi, H., & Hirasawa, S. (2022). *Effect of Hamming Distance on Performance of ECOC with Estimated Binary Classifiers*. 111–116.

Lewis, D. D. (2019). Evaulating and Optimizing Autonomous Text Classification System. *AT & Bell Laboraties*, 246–254.

Li, K., Xie, P., Zhai, J., & Liu, W. (2017). An improved adaboost algorithm for imbalanced data based on weighted KNN. *2017 IEEE 2nd International*

Conference on Big Data Analysis, ICBDA 2017, 30–34. https://doi.org/10.1109/ICBDA.2017.8078849

Mercan, V., Jamil, A., Hameed, A. A., Magsi, I. A., Bazai, S., & Shah, S. A. (2021). Hate Speech and Offensive Language Detection from Social Media. *2021 International Conference on Computing, Electronic and Electrical Engineering, ICE Cube 2021 - Proceedings*. https://doi.org/10.1109/ICECube53880.2021.9628255

Mienye, I. D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access*, *10*(September), 99129–99149. https://doi.org/10.1109/ACCESS.2022.3207287

Parveen, & Singh, A. (2015). Detection of brain tumor in MRI images, using combination of fuzzy c-means and SVM. *2nd International Conference on Signal Processing and Integrated Networks, SPIN 2015*, 98–102. https://doi.org/10.1109/SPIN.2015.7095308

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, *2018-Decem*(Section 4), 6638–6648.

Rodríguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A New classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1619–1630. https://doi.org/10.1109/TPAMI.2006.211

Schapire, R. E. (1997). Using output codes to boost multiclass learning problems. *Proceedings of the Fourteenth International Conference Machine Learning*, *1*, 1–9. http://www.cs.iastate.edu/~jtian/cs573/Papers/Schapire-ICML-97.pdf

Stern, M., Beck, J., & Woolf, B. (1999). Naive Bayes classifiers for user modeling. *Center for Knowledge Communication, …*, *July*, 1–10. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.1676&rep=rep1&type=pdf

Sun, S., & Huang, R. (2010). An adaptive k-nearest neighbor algorithm. *Proceedings - 2010 7th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010*, *1*(Fskd), 91–94. https://doi.org/10.1109/FSKD.2010.5569740

Suzuki, L., Mikawa, K., & Goto, M. (2017). Multi-valued classification of text data

based on an ECOC approach using a ternary orthogonal table. *Industrial Engineering and Management Systems*, *16*(2), 155–164. https://doi.org/10.7232/iems.2017.16.2.155

Tumer, K., & Ghosh, J. (1996). *ERROR CORRELATION AND ERROR REDUCTION IN ENSEMBLE CLASSIFIERS ∗ Kagan Tumer and Joydeep Ghosh*. 1–24.

Webb, G. I. (2017). Encyclopedia of Machine Learning and Data Mining. Içinde *springer* (Sayı April). https://doi.org/10.1007/978-1-4899-7502-7

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). Ensemble learning. Içinde *Data Mining* (ss. 479–501). https://doi.org/10.1016/b978-0-12-804291-5.00012-x

Zhang, F., Zhu, X., Hu, T., Guo, W., Chen, C., & Liu, L. (2016). Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations. *ISPRS International Journal of Geo-Information*, *5*(11). https://doi.org/10.3390/ijgi5110201

Zhang, L., Li, S. Z., Yuan, X., & Xiang, S. (2007). Real-time object classification in video surveillance based on appearance learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2007.383503

Zor, C., Windeatt, T., & Yanikoglu, B. (2011). Bias-variance analysis of ECOC and bagging using neural nets. *Studies in Computational Intelligence*, *373*, 59–73. https://doi.org/10.1007/978-3-642-22910-7_4

https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

https://www.kaggle.com/datasets/extralime/math-lectures

ÖZGEÇMİŞ

Adı Soyadı : Vildan Mercan

Yabancı Dil : İngilizce

EĞİTİM

Lisans	İstanbul üniversitesi/ Fen Fakültesi	Matematik Bölümü (2007-2011)
Lisans	Anadolu Üniversitesi (AÖF)	Sosyoloji (2017-2020)
Yüksek lisans (Tezsiz)	İstanbul Üniversitesi	Pedagojik Formasyon (2010-2011)
Yüksek lisans (Tezli)	İstanbul Sabahattin Zaim Üniversitesi / Mühendislik ve Doğa Bilimleri Fakültesi	Bilgisayar Mühendisliği (2020-2023)

Kurslar : Boğaziçi Üniversitesi İngilizce Kursu (2yıl)

YAYINLAR

Mercan Vildan, Akhtar Jamil, Alaa Ali Hameed, Irfan Ahmed Magsi, Sibghatullah Bazai, and Syed Attique Shah. 2021. *"Hate Speech and Offensive Language Detection from Social Media." 2021 International Conference on Computing, Electronic and Electrical Engineering,ICE Cube 2021 - Proceedings.*

Mercan Vildan , Bedir Sumeyra. 2022. "*A Note on Applications of Error Correcting Output Codes on Detecting Hate Speech in Twitter Data", 4th International Conference on Applied Engineering and Natural Sciences*