T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TWITTER VERİLERİ ÜZERİNDE DUYGU ANALİZİ VE TAHMİNLEMESİ

Yüksek Lisans Tezi

Tezi Hazırlayan
Ali Kağan ÇİFTDEMİR

İstanbul, 2025

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TWITTER VERİLERİ ÜZERİNDE DUYGU ANALİZİ VE TAHMİNLEMESİ

Yüksek Lisans Tezi

Tezi Hazırlayan
Ali Kağan ÇİFTDEMİR

Öğrenci No
2220003040

ORCID ID
0009-0004-6656-5466

Danışman
Doç. Dr. Atınç YILMAZ

İstanbul, 2025

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum **"Makine Öğrenmesi Yöntemleri ile Twitter Verileri Üzerinde Duygu Analizi ve Tahminlemesi"** başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını, patent ve telif haklarını ihlal edici bir davranışımın olmadığını belirtir ve bunu onurumla doğrularım. 29.05.2025

Ali Kağan ÇİFTDEMİR

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ
TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

29/05/2025

Enstitümüz ***Bilgisayar Mühendisliği*** Anabilim Dalı ***Bilgisayar Mühendisliği*** Programı yüksek lisans öğrencilerinden ***2220003040*** numaralı ***Ali Kağan ÇİFTDEMİR***'in *"İstanbul Beykent Üniversitesi Lisansüstü Eğitim – Öğretim Yönetmeliği"*nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği ***"Makine Öğrenmesi Yöntemleri ile Twitter Verileri Üzerinde Duygu Analizi ve Tahminlemesi"*** konulu tezini, Yönetim Kurulumuzun 20/05/2025 tarih ve 2025/22 sayılı toplantısında seçilen ve On-Line toplanan biz jüri üyeleri huzurunda, İstanbul Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 29. maddesinin 3. fıkrası gereğince Zoom programı aracılığıyla on-line olarak aday tarafından savunulmuş ve sonuçta adayın tezi hakkında ***"OYBİRLİĞİ"*** ile ***"KABUL"*** kararı verilmiştir.

İşbu tutanak, 2 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.

DANIŞMAN
Doç. Dr. At*** YI***
(İstanbul Beykent Üniversitesi)

ÜYE
Dr. Öğr. Üyesi Üm*** ÖZ***
(İstanbul Beykent Üniversitesi)

ÜYE
Dr. Öğr. Üyesi So*** SE***
(Kütahya Dumlupınar Üniversitesi)

Adı ve Soyadı : Ali Kağan ÇİFTDEMİR

Danışmanı : Doç. Dr. Atınç YILMAZ

Derecesi ve Tarihi : Yüksek Lisans (Tezli), 2025

Alanı : Bilgisayar Mühendisliği

Anahtar Kelimeler : Veri, Duygu Analizi, Python, Makine Öğrenmesi, Kaggle

ÖZ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE TWITTER VERİLERİ ÜZERİNDE DUYGU ANALİZİ VE TAHMİNLEMESİ

Bu çalışmada, metin tabanlı veriler üzerinde duygu analizi gerçekleştirerek makine öğrenmesi algoritmaları ile tahminleme yapabilen bir uygulama geliştirilmiştir. Uygulama Phyton programlama diliyle geliştirilmiştir. Çalışmada Kaggle platformundan elde edilen ve İngilizce tweetlerden oluşan büyük bir veri setinden rastgele fakat homojen seçilen 10.000 veri kullanılmıştır. Bu verilerin %70'i eğitim, %30'u test amacıyla ayrılmış ve sadece pozitif ve negatif duygu etiketine sahip örnekler modele dahil edilmiştir. Veri ön işleme sürecinde URL, özel karakter, emoji ve gereksiz kelimeler temizlenmiş, ardından kelimeler köklerine indirgenmiştir. Temizlenen veriler TF-IDF yöntemiyle sayısal vektörlere dönüştürülmüş ve n-gram teknikleriyle yapılandırılmıştır. SVM, Random Forest ve Naive Bayes algoritmaları tekil olarak uygulanmış, ayrıca PCA, Kmeans ve Ki-Kare gibi yöntemlerle hibrit modeller geliştirilmiştir. GridSearchCV yöntemi ile en uygun parametreler belirlenerek toplamda 9 farklı model oluşturulmuştur. Model performansları karşılaştırıldığında, hibrit modellerin tekil modellere kıyasla daha yüksek doğruluk oranları sağladığı görülmüştür. Elde edilen sonuçlar, metin madenciliği ve duygu analizi çalışmalarında makine öğrenmesi algoritmalarının etkinliğini göstermiştir. Bu çalışma, farklı modelleme stratejileri sunarak gelecekte yapılacak çalışmalara katkı sağlamayı amaçlamıştır.

Name and Surname : Ali Kağan ÇİFTDEMİR

Supervisor : Assoc. Prof. Dr. Atınç YILMAZ

Degree and Date : Master's (Thesis), 2025

Major : Computer Engineering

Keywords : Data, Sentiment Analysis, Python, Machine Learning, Kaggle

ABSTRACT

SENTIMENT ANALYSIS AND PREDICTION ON TWITTER DATA USING MACHINE LEARNING METHODS

In this study, an application capable of performing sentiment analysis on text-based data and making predictions using machine learning algorithms was developed. The application was implemented using the Python programming language. A randomly but homogeneously selected subset of 10,000 data points was used from a large dataset of English tweets obtained from the Kaggle platform. Of these data, 70% were allocated for training and 30% for testing, including only examples labeled with positive and negative sentiments. During the data preprocessing stage, URLs, special characters, emojis, and unnecessary words were removed, and words were reduced to their root forms. The cleaned data were converted into numerical vectors using the TF-IDF method and structured with n-gram techniques. The algorithms SVM, Random Forest, and Naive Bayes were applied individually, and hybrid models were also developed using methods such as PCA, Kmeans, and Chi-Squared. By utilizing the GridSearchCV method, the optimal parameters were determined, resulting in the creation of a total of nine different models. When the model performances were compared, it was observed that the hybrid models achieved higher accuracy rates compared to the individual models. The results demonstrated the effectiveness of machine learning algorithms in text mining and sentiment analysis. This study aims to contribute to future research by presenting various modeling strategies.

İÇİNDEKİLER

Sayfa No.

ÖZ

ABSTRACT

ŞEKİLLER LİSTESİ iii

KISALTMALAR iv

SÖZLÜK v

GİRİŞ 1

1. LİTERATÜR TARAMASI

2. YAPAY ZEKA

2.1. Yapay Sinir Ağlarının Temel Bileşenleri 17

2.2. Yapay Sinir Ağ Modelleri 18

2.2.1. Tek Katmanlı Yapay Sinir Ağları 18

2.2.2. Çok Katmanlı Yapay Sinir Ağları 19

2.2.3. İleri Beslemeli Yapay Sinir Ağları 20

2.2.4. Geri Beslemeli Yapay Sinir Ağları 21

3. MAKİNE ÖĞRENMESİ

3.1. Denetimli Öğrenme 22

3.1.1. Destek Vektör Makineleri (SVM) 23

3.1.2. Naive Bayes 24

3.1.3. Rastgele Orman (Random Forest) 27

3.1.4. Özellik Seçimi (Feature Selection) 29

3.1.4.1. Ki-Kare Testi (Chi-Squared Test) 33

3.2. Denetimsiz Öğrenme 36

3.2.1. Temel Bileşenler Analizi (PCA) 37

3.2.2. Kmeans 40

3.3. Yarı Denetimli Öğrenme 41

3.4. Takviyeli Öğrenme 41

4. DERİN ÖĞRENME

5. DUYGU ANALİZİ

6. PYTHON

6.1. Pandas 55

6.2. Numpy 58

6.3. NLTK (Natural Language Toolkit) 61

6.4. Scikit-Learn (SKLEARN) 62

7. UYGULAMA

SONUÇ 74

KAYNAKÇA 86

ŞEKİLLER LİSTESİ

Sayfa No.

Şekil 1. Zekâ, Makine Öğrenmesi ve Derin Öğrenme Arasındaki İlişki 15

Şekil 2. Biyolojik Sinir Hücresi 16

Şekil 3. Yapay Sinir Hücresi 17

Şekil 4. Tek Katmanlı Yapay Sinir Ağı Modeli 18

Şekil 5. Çok Katmanlı Yapay Sinir Ağı Modeli 19

Şekil 6. İleri Beslemeli Yapay Sinir Ağı Modeli 20

Şekil 7. Geri Beslemeli Yapay Sinir Ağı Modeli 21

Şekil 8. SVM Destek Vektörleri ve Hiper Düzlemi 24

Şekil 9. Naive Bayes Teoremi Formülü 25

Şekil 10. Ki-Kare Testi (Chi-Squared) Yöntemi Formülü 34

Şekil 11. Bağımsızlık ve Homojenlik Formülü 34

Şekil 12. SVM Modelinin Confusion Matrix Grafiği 74

Şekil 13. Naive Bayes Modelinin Confusion Matrix Grafiği 75

Şekil 14. Random Forest Modelinin Confusion Matrix Grafiği 76

Şekil 15. PCA + SVM Modelinin Confusion Matrix Grafiği 76

Şekil 16. PCA + Random Forest Modelinin Confusion Matrix Grafiği 77

Şekil 17. Ki-Kare + SVM Modelinin Confusion Matrix Grafiği 78

Şekil 18. Ki-Kare + Random Forest Modelinin Confusion Matrix Grafiği 79

Şekil 19. Kmeans + SVM Modelinin Confusion Matrix Grafiği 80

Şekil 20. Kmeans + Random Forest Modelinin Confusion Matrix Grafiği 81

KISALTMALAR

AI : Artificial Intelligence (Yapay Zeka)

GUI : Graphical User Interface (Grafik Kullanıcı Arayüzü)

ML : Machine Learning (Makine Öğrenmesi)

NB : Naive Bayes (Naive Bayes)

NLP : Natural Language Processing (Doğal Dil İşleme)

NLTK : Natural Language Toolkit (Doğal Dil Araç Seti)

PCA : Principal Component Analysis (Temel Bileşen Analizi)

RF : Random Forest (Rastgele Orman)

SVM : Support Vector Machine (Destek Vektör Makinesi)

SÖZLÜK

Derin Öğrenme: Çok katmanlı yapay sinir ağlarını kullanarak karmaşık veri örüntülerini yüksek doğrulukla öğrenmeyi hedefleyen bir makine öğrenmesi alt alanıdır. Özellikle görüntü tanıma, doğal dil işleme gibi alanlarda başarılı sonuçlar vermektedir.

Duygu Analizi: Bir metnin içerdiği duygusal tonu (olumlu, olumsuz, nötr) belirlemeye yönelik doğal dil işleme tekniklerini kullanan bir analiz yöntemidir. Genellikle müşteri geri bildirimleri ve sosyal medya verileri gibi metin kaynakları üzerinde uygulanır.

Doğal Dil İşleme: Bilgisayarların insan dilini (metin ve konuşma) anlamasını, yorumlamasını ve üretmesini amaçlayan yapay zeka alanıdır.

Makine Öğrenmesi: Bilgisayarların açıkça programlanmadan veri yoluyla öğrenmesini ve deneyim kazanmasını sağlayan bir yapay zeka alt alanıdır. Algoritmalar aracılığıyla verideki örüntüleri tanıyarak tahminler veya kararlar üretir.

Yapay Sinir Ağı: İnsan beyninin yapısından esinlenerek geliştirilmiş, birbirine bağlı düğümlerden (nöronlar) oluşan ve karmaşık örüntüleri tanıma, sınıflandırma gibi görevlerde kullanılan bir makine öğrenmesi alt alanıdır.

Yapay Zeka: İnsan bilişsel yeteneklerini (öğrenme, problem çözme, karar verme vb.) taklit edebilen bilgisayar sistemleri ve yazılımlarıdır.

GİRİŞ

Twitter, 2006 yılında kullanıma sunulmuş ve 2017 yılından itibaren kullanıcılarına herhangi bir konuda görüşlerini en fazla 280 karakter uzunluğunda paylaşma imkânı tanıyan, günde 500 milyondan fazla paylaşım yapılan dünyaca popüler bir mikro blog platformudur. Tweetler, zaman damgasıyla birlikte zaman akışı üzerinde sunulmakta ve kullanıcılar doğrudan mesajlar aracılığıyla ya da hesap adlarının geçtiği içeriklere yanıt vererek birbirleriyle etkileşim kurabilmektedir. Bu etkileşim, üç temel sembolle gerçekleştirilmektedir: "@" sembolü kullanıcı adının başında yer alır, paylaşımı yeniden yayınlamak için "RT" kullanılır, ve "#", yani hashtag, bir kelime ya da cümleyle konuya katılım sağlar.

Twitter'ın kullanımındaki artış, bu platformun özellikle kalite yönetimi ve algı yönetimi gibi alanlarda "sosyal bir megafon" olarak görülmesine neden olmuştur. Bu nedenle, Twitter gibi sosyal medya araçlarını stratejik şekilde kullanan işletmeler bu platformlardan olumlu şekilde yararlanabilir. Ayrıca, Twitter verileri müşteri geri bildirimlerinin toplanması ve duygu analizi yapılması açısından zengin bir kaynak olarak kabul edilmektedir.

Günümüzde internetin hızlı gelişimi, onu insanlar için su ve hava kadar vazgeçilmez bir ihtiyaç haline getirmiştir. İnsanlar, kişisel görüşlerini ve ilgi alanlarını paylaşma isteğiyle sosyal medyayı büyük bir bilgi kaynağına dönüştürmüştür. Ancak, sosyal medya verileri ham haliyle çalışılması zor verilerden oluşmaktadır. Bu verilerde, sıkça hatalı yazılmış kelimeler, kısaltmalar ve sosyal medyaya özgü jargon kelimeler yer alır. Bu sebeple, bu tür verilerin işlenmesi için doğal dil işleme teknikleri kullanılarak filtrelenmesi ve düzenlenmesi gerekmektedir. Twitter'dan toplanan veriler ile doğal dil işleme, veri madenciliği, metin madenciliği, duygu analizi alanlarında pek çok çalışma yapılmıştır. Birbirinden farklı bir çok teknoloji kullanılmıştır.

Bu yöntemler kullanılmadan önce elde etmemiz gereken en önemli şey veridir. Veri, herhangi bir olay, durum veya olguya ilişkin toplanmış, işlenmemiş ve organize edilmemiş ham bilgilerdir. Tek başına anlam ifade etmeyebilir ve genellikle düzenlenmesi, işlenmesi veya yorumlanması gerekir. Veriler genellikle sayılar, kelimeler, ölçümler, gözlemler veya semboller şeklinde ortaya çıkar.

Bilgi ise, verinin işlenip anlamlı hale getirilmiş halidir. Veriler analiz edilip belirli bir bağlamda değerlendirildiğinde bilgiye dönüşür. Bilgi, belirli bir olay veya durumu daha iyi anlamak ve karar vermek için kullanılır. Sonuç olarak, veri ham bir kaynakken, bilgi bu kaynağın işlenmiş ve anlamlandırılmış versiyonudur. Verilerin bilgiye dönüştürülmesi, doğru kararlar almak ve problemleri çözmek açısından büyük önem taşır.

Duygu analizi (Sentiment Analysis), bir metnin duygusal tonunu tespit etmeye yönelik doğal dil işleme (NLP) tekniklerini kullanan bir analiz yöntemidir. Bu analiz, metinlerin olumlu, olumsuz ya da nötr duygu taşıyıp taşımadığını belirlemek için kullanılır. Genellikle müşteri geri bildirimleri, sosyal medya gönderileri, incelemeler ve anket verileri gibi metin kaynakları üzerinde yapılır. Örnek olarak; müşteri yorumları veya şikayetleri analiz edilerek, müşteri memnuniyeti ve markaya olan genel yaklaşım ölçülebilir. Sosyal medya gönderileri analiz edilerek bir marka, ürün veya etkinlik hakkındaki genel duygusal eğilimler belirlenebilir. Ürün ve hizmetler hakkında tüketici görüşleri analiz edilerek, yeni stratejiler geliştirilebilir. Seçim kampanyaları veya liderler hakkında halkın genel görüşleri belirlenebilir.

Twitter verilerine erişmenin birden çok yöntemi mevcuttur. Twitter, geliştiriciler için sunduğu Twitter API ile kullanıcılara tweetler, kullanıcı bilgileri, hashtag'ler ve daha birçok veri kaynağına erişim sağlar. Ya da Twitter API kullanarak aldığınız verileri veya Kaggle'dan indirilen verileri doğrudan bir dosyaya kaydedip, projede bu dosyadan okumak mümkündür.

Verilere eriştikten sonra bu projeyi modelleyebilmemiz için bir programlama diline ihtiyaç vardır. Programlama dilleri, modern teknolojinin temel taşlarını oluşturan, bilgisayarlarla insan arasındaki iletişimi sağlayan araçlardır. Farklı görevler ve ihtiyaçlar için geliştirilmiş çok sayıda programlama dili bulunmaktadır. Bu diller, yazılım geliştiricilere, bilgisayarların anlamlandırabileceği komutlar yazma imkanı tanır.

İlk programlama dilleri, donanımın sınırlarına göre oluşturulmuş, makine kodlarına dayalı dillerdi. Zamanla, daha soyut ve insana yakın diller gelişti. C, Pascal ve Fortran gibi diller, düşük seviye donanım yönetimi sağlarken, Python, Java ve JavaScript gibi modern diller, kullanıcı dostu yapıları ve zengin kütüphaneleriyle geniş bir kullanım alanına sahiptir.

Programlama dillerinin gelişimiyle birlikte, her bir dilin belirli bir amacı olduğu ortaya çıktı. Örneğin, JavaScript web geliştirmede yaygın olarak kullanılırken, Python veri bilimi, yapay zeka ve otomasyon alanlarında tercih edilmektedir. C++ ve C# ise oyun ve uygulama geliştirme gibi daha performans odaklı alanlarda öne çıkar.

Günümüzde programlama dillerinin en önemli özelliklerinden biri, çok platformlu ve esnek yapıya sahip olmalarıdır. Birçok dil, farklı işletim sistemlerinde çalışabilir ve çeşitli cihazlar arasında entegrasyonu kolaylaştırır. Aynı zamanda, modern programlama dilleri açık kaynak kodlu kütüphaneler ve çerçeveler sunarak, yazılımcıların işlerini hızlandırmalarına ve daha yenilikçi çözümler üretmelerine imkan tanır.

Sonuç olarak, programlama dilleri, teknoloji dünyasının sürekli gelişimini yönlendiren kritik araçlardır. Her bir dilin kendine özgü güçlü yanları ve kullanım alanları bulunur, bu nedenle projeye ve hedefe uygun dilin seçilmesi yazılım geliştirme sürecinin en önemli adımlarından biridir. Bu bağlamda, duygu analizi gibi doğal dil işleme (NLP) uygulamaları, projelerde öne çıkan alanlardan biridir. Duygu analizi, bir metnin içindeki duygusal tonları ve eğilimleri tespit etmeye çalışan bir tekniktir ve genellikle metnin olumlu, olumsuz veya nötr duygular içerip içermediğini belirlemek için kullanılır. Sosyal medya, müşteri geri bildirimleri, ürün incelemeleri ve haber makaleleri gibi çeşitli metin kaynakları üzerinde yapılan analizlerde sıkça tercih edilir.

Genelde doğal dil işleme, veri madenciliği, metin madenciliği, duygu analizi alanlarında Python programlama dili kullanılmaktadır. Bu yapılan çalışmada da en yaygın kullanılan programlama dili olan Python kullanılmıştır. Yapılacak çalışmada bir çok kütüphane projeye entegre edilmiştir. Bunlar arasında Pandas, NLTK, Numpy, Sklearn bulunmaktadır. Bu kütüphaneler yardımıyla, Kaggle platformu üzerinden elde edilen Twitter verileri üzerinde duygu analizi yapılmıştır.

Excel formatındaki veriler projeye entegre edildikten sonra, cümleler üzerinde özel karakterlerin temizlenmesi, stop words'lerin çıkarılması ve cümlelerin kelimelere ayrılması işlemleri gerçekleştirilmiştir. Ardından, kelimeler köklerine ayrılmıştır. Modele verilen veri setleri etiketlidir yani her giriş değerine karşılık bir çıkış değer karşısındaki kolonda mevcuttur. Veri setindeki cümlelerin duygusal analizleri yapılmış, bu veriler makine öğrenme modeline öğretilmiştir. Son aşamada ise,

makineye verilen test verileri üzerinde, önceden öğrenilen veriler kullanılarak duygusal tahminlemeler yapılmıştır.

Makine öğrenmesi (ML), bilgisayarların veri yoluyla öğrenmesini ve deneyim kazanmalarını sağlayan bir yapay zeka (AI) alt alanıdır. İnsan müdahalesine ihtiyaç duymadan, makinelerin belirli görevleri otomatik olarak iyileştirmesine olanak tanır. Temel olarak, makine öğrenmesi algoritmaları, büyük miktarda veriyi analiz eder, örüntüleri tanır ve bu verilere dayanarak gelecekteki tahminleri yapar. Temel makine öğrenmesi yöntemleri şunlardır:

Denetimli Öğrenme (Supervised Learning)**:** Bu yöntemde, model, etiketlenmiş veri kullanılarak eğitilir. Yani, her veri girişi için doğru çıktı (etiket) bilinir. Model, girdi ile çıktı arasındaki ilişkiyi öğrenerek, yeni verilere dayalı doğru tahminler yapmayı hedefler.

Denetimsiz Öğrenme (Unsupervised Learning)**:** Bu türde, model etiketlenmemiş verilerle çalışır. Model, verilerdeki örüntüleri veya yapıları kendi başına keşfetmeye çalışır.

Yarı Denetimli Öğrenme (Semi-supervised Learning)**:** Bu yöntemde, bazı verilerin giriş bilgileri için doğru çıkış bilgilerini bilemeyebiliriz. Çünkü bazı veriler gürültülü, eksik veya boş olabilir. Dolayısıyla bu öğrenme yönteminde ilk önce denetimsiz öğrenme yapılarak verilere etiket bilgisi kazandırılır. Daha sonra ise etiket bilgisi kazandırılmış verilere Denetimli öğrenme yapılarak sonuçlar ortaya konur.

Makine öğrenmesi uygulamaları, işletmelerin karar alma süreçlerini geliştirmelerine yardımcı olabilir. Örneğin, tahmin modelleri, müşteri davranışlarını öngörmek, satışları artırmak veya riskleri azaltmak için kullanılabilir. Ayrıca, doğal dil işleme ve duygu analizi gibi alanlarda da makine öğrenmesi teknikleri önemli rol oynar. Bu teknikler, metinleri analiz ederek duygusal tonları belirleyebilir ve müşterilerin görüşlerini daha iyi anlamak için kullanılır.

Veri bilimi, makine öğrenmesi ve yapay zeka alanları, veri işleme ve analiz yeteneklerini sürekli olarak geliştirir. Bu alanlardaki yenilikler, büyük veri kümelerinin anlamlı bilgiye dönüştürülmesini sağlar. Bu süreç, veri toplama, veri temizleme, özellik mühendisliği, model seçimi ve değerlendirme aşamalarını içerir. Veri bilimciler, bu aşamalarda çeşitli araçlar ve teknikler kullanarak veri analizini optimize eder ve iş kararlarına katkıda bulunur.

Bu çalışmada bir çok Makine Öğrenmesi yöntemi ile tahminleme yapılarak sonuçları elde edilmiştir. Tek başına Makine Öğrenmesinin sınıflandırma yöntemlerinden bazılarıyla modelleme yapılırken, hibrit yaklaşımlar da ele alınmıştır. Hibrit modellerde sadece sınıflandırma yöntemleriyle yapılan modellerden daha iyi doğru tahminleme sonuçları elde edilmeye çalışılmıştır. Hibrit modellerde farklı yaklaşımlar ele alınmıştır. Örneğin veri setindeki veriler vektörize edildikten sonra bir kümeleme yöntemi olan PCA yöntemi yardımıyla boyutu indirgenmiştir. Ardından kümeleme yönteminden çıkan sonuçlar sınıflandırma yöntemlerine gönderilmiş ve sonuçlar karşılaştırılmıştır.

Bu çalışmada şu sorulara yanıt aranacaktır, "Twitter platformundan elde edilen metin verileri üzerinde sadece sınıflandırma yöntemleriyle yapılan duygu analizi tahminlemesi, farklı makine öğrenmesi yöntemleri ve hibrit yaklaşımlar kullanılarak daha iyi tahminleme sonuçları elde edilebilir mi, eğer elde edilebilirse en yüksek doğruluk hangi yöntemle elde edilebilir?" şeklinde belirlenmiştir.

Bu çalışmanın amacı, Twitter'dan elde edilen metin verileri üzerinde doğal dil işleme ve makine öğrenmesi tekniklerini kullanarak duygu analizi gerçekleştirmek ve farklı sınıflandırma yöntemleri ile hibrit modelleme yaklaşımlarının performanslarını karşılaştırarak en etkili yöntemi ortaya koymaktır. Bu kapsamda, verilerin ön işleme, özellik çıkarımı, boyut indirgeme, özellik seçimi ve sınıflandırma aşamaları ayrıntılı olarak ele alınmıştır.

Bu çalışma kapsamında test edilen hipotezler şunlardır:

- Farklı makine öğrenmesi algoritmaları arasında duygu analizi doğruluğu bakımından anlamlı bir fark vardır.
- Hibrit yaklaşımlar (kümeleme ve sınıflandırma yöntemlerinin birleşimi veya özellik seçimi ve sınıflandırma yöntemlerinin birleşimi), yalnızca sınıflandırma yöntemlerinden daha yüksek doğruluk sağlar.
- Veri ön işleme adımları (temizleme, kök bulma, stop-words çıkarma vb.), modelin genel başarımını artırır.

Bu çalışmada, Twitter'dan elde edilen veri setinin, kullanıcıların gerçek duygu ve düşüncelerini yansıttığı kabul edilmiş ve etiketlenmiş veri setinin doğruluğu ve

güvenilirliği kabul edilmiştir. Aynı zamanda bu çalışma, yalnızca Kaggle üzerinden elde edilen belirli bir Twitter veri seti ile sınırlandırılmıştır. Sadece İngilizce veriler üzerinde çalışılmıştır ve diğer diller dikkate alınmamıştır.

Sonuç olarak, veri bilimi ve makine öğrenmesi, modern iş dünyasında rekabet avantajı sağlamak için kritik araçlardır. İşletmeler, bu teknolojileri kullanarak veri odaklı kararlar alabilir, operasyonel verimliliği artırabilir ve müşteri memnuniyetini iyileştirebilir. Doğru veri yönetimi ve analitik yaklaşımlar, stratejik hedeflere ulaşmada önemli bir rol oynar ve teknolojinin sunduğu fırsatlardan en iyi şekilde yararlanmayı sağlar.

1. LİTERATÜR TARAMASI

Literatür taramasında Twitter verileri ile doğal dil işleme, duygu analizi gibi bir çok konuda örnekler mevcuttur. Bu çalışmanın ilk aşamasında Twitter üzerinden Twitter API üzerinden çekilen verilerle ya da Kaggle üzerinden alınan veriler üzerinde çeşitli veri temizleme işlemleri yapılarak verinin duygu analizi yapılabilir hale getirilmesini sağlamaktır. Sonrasında ise duygu analizi yapılmaldırı. Bu işlem için farklı yöntemler mevcuttur. Duygu analizi de yapıldıktan sonra artık verilerin gerçek duyguları bilindiği için Makine Öğrenmesi yöntemleri yardımıyla test verileri üzerinden tahminleme yapılabilir. Son aşama olarak elde bulunan verinin etiketli olup olmamasına göre ise yöntemler belirlenebilir.

İlhan ve Sağaltıcı, Twitter üzerinde yapılan duygu analizi uygulamalarını ele alarak, büyük ölçekli sosyal medya verilerinden anlamlı bilgiler elde etmeyi amaçlayan bir çalışma gerçekleştirmiştir. Çalışmada 1.578.627 adet etiketlenmiş tweet üzerinde Naïve Bayes ve Destek Vektör Makineleri (SVM) algoritmaları kullanılarak, tweetlerin olumlu veya olumsuz olarak sınıflandırılması hedeflenmiştir. Özellikle SVM algoritmasıyla %64 doğruluk oranına ulaşılırken, Naïve Bayes algoritmasıyla bu oran %42 seviyesinde kalmıştır. Bu çalışma, metin ön işleme adımlarına (tokenizasyon, gereksiz karakterlerin temizlenmesi, kök bulma işlemleri gibi) büyük önem vermiştir. Ayrıca, SentiWordNet sözlüğünden faydalanılarak kelimelerin duygu değerleri hesaplanmış ve N-gram (Unigram ve Bigram) yöntemleri kullanılarak özellik çıkarımı gerçekleştirilmiştir. Ancak yöntemlerin doğruluk oranlarının sınırlı kalması, özellikle VADER gibi hazır araçların %27 gibi düşük başarı göstermesi, sistemlerin kısa, belirsiz ve imla hatası içeren sosyal medya metinlerinde yeterince başarılı olamadığını ortaya koymaktadır. Literatürde benzer çalışmaların çoğunda yalnızca pozitif ve negatif ayrımı yapılmakta, nötr duygular veya daha detaylı duygu türleri göz ardı edilmektedir. Ayrıca, kelime tabanlı yöntemler bağlamsal analiz eksikliği nedeniyle karmaşık cümle yapılarını yeterince çözümleyememektedir. İlhan ve Sağaltıcı'nın çalışmasında da, bu sınırlamalar açıkça gözlemlenmiştir. Söz konusu çalışma, mevcut yöntemlerin büyük veri setlerinde uygulanabilirliğini göstermesi bakımından önemli olmakla birlikte, daha derin bağlam analizi veya çok katmanlı duygu sınıflandırması gibi gelişmiş yaklaşımlara yer vermemiştir.

Bu bağlamda hazırlanan tez çalışması, literatürdeki bu boşlukları doldurmayı amaçlamaktadır. Öncelikle yalnızca kelime frekansına dayalı geleneksel yöntemler yerine, derin öğrenme tabanlı dil modelleri kullanılarak bağlam dikkate alınmakta; kısa, dil bilgisi hataları içeren veya çok anlamlı ifadelerin yer aldığı sosyal medya metinlerinin analizinde daha yüksek başarı hedeflenmektedir. Ayrıca, yalnızca ikili (pozitif/negatif) değil, çok sınıflı (öfke, mutluluk, üzüntü, korku gibi) duygu kategorileri üzerinden sınıflandırma yapılmasıyla literatüre yenilikçi bir katkı sağlanmaktadır. Bu sayede, sosyal medya verilerinden daha anlamlı ve derinlemesine içgörüler elde edilmesi mümkün hale getirilmektedir (İlhan ve Sağaltıcı, 2020, s. 147).

Demir vd. (2019, s. 58) tarafından yürütülen çalışma, Türkçe metinlerde duygu analizine yönelik sözlük tabanlı bir yaklaşım geliştirmiştir. Çalışmada, sosyal medya gönderileri, kitap yorumları ve müşteri geri bildirimleri gibi Türkçe metinlerin analizine olanak sağlayan bir sistem önerilmiştir. Sistem, üç farklı Türkçe kelime sözlüğü kullanarak metinleri analiz etmekte ve Bayes sınıflandırıcı yardımıyla pozitif ya da negatif duygu yönelimine göre sınıflandırmaktadır. Deneyler sonucunda %77 ile %83 arasında değişen doğruluk oranları elde edilmiştir. Çalışma, özellikle Türkçe metinler üzerinde gerçekleştirilen sözlük tabanlı yaklaşımların etkinliğini göstermesi bakımından literatürde önemli bir yer tutmaktadır. Ancak mevcut çalışmanın bazı sınırlılıkları göze çarpmaktadır. Öncelikle, analiz yalnızca sözlük tabanlı yöntemle sınırlandırılmış, bağlamı dikkate alan modern derin öğrenme modellerine yer verilmemiştir. Ayrıca, metinlerdeki imla hataları, kinaye, çok anlamlı ifadeler gibi dilin doğal yapısından kaynaklanan karmaşıklıklar tam anlamıyla ele alınamamış ve analiz doğruluğunu olumsuz yönde etkilemiştir. Çalışmada kullanılan veri setlerinin çoğunluğunun kullanıcı yorumlarından oluşması ve sosyal medya gibi dinamik ve kısa metinlere yönelik özel bir derinlemesine değerlendirme yapılmamış olması, yöntemlerin genel uygulanabilirliğini sınırlamaktadır. Çalışmalar arasında kullanılan yöntemlerin başarı düzeyleri karşılaştırıldığında, Demir ve arkadaşlarının sözlük tabanlı yaklaşımı, küçük ve orta ölçekli veri setlerinde belirli bir doğruluk sağlarken, daha büyük ve karmaşık metinlerde sınırlı kalmıştır. Diğer yandan, literatürdeki birçok çağdaş çalışma, derin öğrenme tabanlı modellerin özellikle BERT gibi bağlamsal dil modellerinin daha yüksek başarı oranları sunduğunu göstermektedir.

Buna karşın, bu çalışma, yalnızca kelime düzeyinde işlem yapmakta ve dilin bağlamsal yapısını tam olarak değerlendirememektedir. Bu noktada, mevcut tez çalışması literatürdeki bu boşluğu doldurmayı amaçlamaktadır. Çalışmada, sadece kelime frekansına dayalı değil, bağlamı dikkate alan ileri düzey derin öğrenme yöntemleri kullanılmakta; Türkçe doğal dil işleme alanında eksikliği hissedilen çok boyutlu duygu analizine odaklanılmaktadır. Ayrıca, sosyal medya gibi dinamik platformlardan elde edilen kısa metinlerde, kullanıcı etkileşimleri ve içerik çeşitliliği de dikkate alınarak daha kapsamlı bir analiz yöntemi geliştirilmektedir. Bu sayede hem yüksek doğruluk oranlarına ulaşılması hem de Türkçe metinlerde karşılaşılan dilsel zorlukların daha etkili yönetilmesi hedeflenmektedir (Demir vd., 2019, s. 58).

Kandıran vd. (2022, s. 228) tarafından gerçekleştirilen çalışma, Covid-19 pandemisi sürecinde Türkiye'de uzaktan eğitime ilişkin kamuoyunun sosyal medya üzerindeki tepkilerini inceleyen önemli araştırmalardan biridir. Çalışmada, 16 Mart 2020 - 17 Mayıs 2021 tarihleri arasında Twitter'da eğitimle ilgili öne çıkan 28 etiket üzerinden paylaşılan 8545 Türkçe tweet analiz edilmiştir. Bu analizde Python'un Tweepy modülü aracılığıyla Twitter API kullanılmış, metin ön işleme adımlarının ardından Türkçe BERT modeliyle duygu analizi gerçekleştirilmiştir. Çalışma, özellikle vaka artışlarıyla uzaktan eğitime ilişkin olumlu ya da olumsuz duygu yoğunluklarının nasıl değiştiğini ortaya koymuş ve sosyal medya verilerinden elde edilen bulgularla kamuoyunun tutumlarını anlamaya çalışmıştır. Literatürdeki diğer çalışmalar genel olarak uzaktan eğitimin öğrenci ya da öğretmen deneyimleri üzerinden değerlendirilmiş, sosyal medya gibi dinamik platformlardan elde edilen büyük veri setleriyle yapılan analizlere sınırlı sayıda yer verilmiştir. Kandıran ve arkadaşlarının çalışması bu açıdan literatürdeki önemli bir boşluğu doldurmakta ve Türkçe sosyal medya verilerini sistematik olarak analiz eden öncü örneklerden biri olmaktadır. Ancak çalışmada kullanılan yöntemin sınırlılığı nedeniyle analiz sadece olumlu ve olumsuz sınıflandırmasına indirgenmiş, duygu çeşitliliği derinlemesine ele alınmamıştır. Ayrıca, sosyal medya kullanıcılarının hangi demografik özelliklere sahip olduğu, tweetlerin hangi kullanıcı gruplarından geldiği gibi detaylar analiz edilmemiştir. Bu durum, çalışmanın bulgularının toplum geneline ne ölçüde genellenebileceği konusunda bazı belirsizlikler yaratmaktadır.

Diğer taraftan, kullanılan BERT modeli Türkçe için uyarlanmış olmasına rağmen, sosyal medya dilinin yapay zeka modelleri için hâlen bazı zorluklar içerdiği bilinmektedir. Modelin sınıflandırma başarısı, metinlerin anlam derinliğini tam olarak yansıtamayabilir. Ayrıca, içerik analizi gibi yöntemlerin kullanılmaması nedeniyle paylaşımların hangi argümanları içerdiği detaylıca ortaya konmamıştır. Çalışmada ayrıca, analiz edilen tweetlerin yalnızca duygusal yönelimi belirlenmiş, bu duyguların sebepleri veya içerikleri hakkında ayrıntılı yorum yapılmamıştır. Bu noktada, mevcut tez çalışmasının farkı; sadece duygu analizine değil, aynı zamanda içerik analizine, argüman tespitine ve kullanıcı etkileşim dinamiklerine de odaklanarak daha kapsamlı bir çözümleme sunmasıdır. Ayrıca, sadece olumlu/olumsuz kutupluluğa dayalı sınıflandırmalar yerine, çok boyutlu duygu analiz modelleri ve bağlamsal içerik çözümleriyle daha derinlemesine bir değerlendirme yapmayı hedeflemektedir. Bu sayede, sosyal medya üzerinden kamuoyu eğilimlerinin, toplumsal hassasiyetlerin ve kullanıcıların ifade ettikleri sorunların bütüncül biçimde ortaya konması amaçlanmaktadır. Bu yaklaşım, mevcut literatürde tespit edilen boşluğu doldurmakta ve sosyal medya temelli eğitim araştırmalarına yeni bir katkı sağlamaktadır (Kandıran vd., 2022, s. 228).

Kılıç vd. (2020, s. 131) tarafından yürütülen çalışmada, Türkiye'de firmalar tarafından kullanılan farklı Kara Cuma etiketleri (örn. #efsaneCuma, #bereketliCuma) ele alınmış ve bu etiketler, tweet istatistikleri ve duygu analizi kullanılarak çeşitli boyutlarda sıralanmıştır. Çalışmada, üç hafta boyunca atılan 2038 tweet analiz edilmiş, toplam tweet sayısı, etkileşim oranları ve kullanıcı bilgileri dikkate alınarak markalar için etiket tercih rehberi sunulmuştur. Etiketler hem sayısal veriler hem de duygusal içerikler bakımından sıralanmış, en fazla kullanılan etiketin #efsaneCuma olduğu ortaya konmuştur. Ancak duygu analizi sonuçlarına göre, en olumlu içeriklerin #süperCuma etiketi altında toplandığı belirlenmiştir. Bu yönüyle çalışma, markaların sadece görünürlük değil, kullanıcı algısına uygun etiket seçimi yapmaları gerektiğini vurgulamaktadır. Mevcut çalışmanın dikkat çekici katkıları olmasına rağmen, bazı sınırlılıkları olduğu da gözlemlenmiştir. Öncelikle, analiz yalnızca Türkiye'de kullanılan belirli Kara Cuma etiketleriyle sınırlıdır ve farklı kampanya türleri veya sosyal medya platformları kapsam dışı bırakılmıştır.

Ayrıca, duygu analizinde yalnızca pozitiflik ve negatiflik düzeyi dikkate alınmış, duyguların çeşitliliği (örneğin, öfke, şaşkınlık, heyecan gibi) detaylandırılmamıştır. Kullanılan yöntemlerin başarı düzeyleri genel hatlarıyla ortaya konmuş, ancak farklı duygu analiz araçları ya da makine öğrenmesi tekniklerinin karşılaştırılmasına yönelik daha derin bir metodolojik analiz yapılmamıştır. Bu nedenle, çalışmanın sonuçlarının genellenebilirliği sınırlı kalmıştır. Literatürde dikkat çeken bir diğer eksiklik, sosyal medya kullanıcılarının demografik özellikleri veya kültürel arka planlarının analiz edilmemiş olmasıdır. Türkiye gibi kültürel ve dini hassasiyetlerin yüksek olduğu toplumlarda, kullanılan etiketlerin yalnızca istatistiksel başarıları değil, toplumsal algı üzerindeki etkileri de önem arz etmektedir. Kılıç ve arkadaşlarının çalışması bu yönde bir tespit sunsa da, bu durumun nicel analizlerle desteklenmediği görülmektedir. Ayrıca, çalışmanın yalnızca 2018 yılı verilerini kapsaması, dinamik sosyal medya trendlerinin uzun dönemli etkilerini gözlemlemeye olanak tanımamıştır. Bu tez çalışması, mevcut literatürdeki bu boşlukları doldurmayı amaçlamakta ve çoklu sosyal medya platformlarından elde edilecek daha geniş kapsamlı veri kümeleriyle analizleri derinleştirmeyi hedeflemektedir. Ayrıca, yalnızca etiketlerin popülerliği değil, kullanıcı profilleri, demografik eğilimler, zaman serisi analizi ve duygu çeşitliliği gibi unsurlar da kapsamlı bir şekilde değerlendirilecektir. Böylece, yalnızca anlık popülerlik değil, markaların uzun vadeli iletişim stratejilerine yön verecek nitelikli içgörüler sunulması hedeflenmektedir. Bu yönleriyle, mevcut çalışma yöntemsel derinliği artırarak ve literatürdeki sınırlı kapsamı genişleterek önemli bir boşluğu doldurmayı amaçlamaktadır (Kılıç vd., 2020, s. 131).

Tuzcu tarafından gerçekleştirilen çalışmada, çevrimiçi bir kitap satış sitesinden elde edilen kullanıcı yorumları üzerinden duygu analizi yapılmıştır. Araştırmada Python programlama diliyle Çok Katmanlı Algılayıcı (MLP) algoritması kullanılmış, ardından RapidMiner yazılımı ile Naive Bayes (NB), Destek Vektör Makineleri (DVM) ve Lojistik Regresyon (LR) algoritmalarının performansları karşılaştırılmıştır. Sonuçlar, MLP algoritmasının %89 doğruluk oranı ile en yüksek başarıyı sağladığını göstermiştir. Bu çalışma, Türkçe dilinde gerçekleştirilen duygu analizi araştırmaları arasında öne çıkan örneklerden biri olarak dikkat çekmektedir. Bununla birlikte, Tuzcu'nun çalışması sınıflandırma performansı açısından önemli sonuçlar üretse de, literatürde yer alan diğer çalışmalara kıyasla bazı sınırlılıkları da barındırmaktadır.

Öncelikle, çalışma yalnızca tek bir veri kaynağına (kitap satış sitesi) ve iki sınıflı (pozitif/negatif) duygu yapısına odaklanmıştır. Ayrıca, kullanılan yöntemlerin neden daha başarılı veya başarısız olduğuna dair metodolojik derinlik sınırlıdır. Örneğin, hangi parametrelerin veya model yapılandırmalarının sonuçları etkilediğine ilişkin ayrıntılı bir çözümleme sunulmamıştır. Literatürdeki diğer çalışmalarla yöntem kıyaslamasına daha fazla yer verilmiş olsaydı, yöntemin genel geçerliliği daha net ortaya konabilirdi. Ayrıca, literatürde Türkçe diline özgü duygu analizi araçlarının sınırlı olması, bu alanda hâlâ büyük bir boşluk bulunduğunu göstermektedir. Mevcut çalışmaların çoğu İngilizce dilinde yürütülmekte ve Türkçe için optimize edilmiş duygu sözlükleri veya algoritmalar sınırlı sayıdadır. Tuzcu'nun çalışması bu noktada önemli bir adım atmış olsa da, veri çeşitliliği, farklı dil modellerinin kıyaslanması ve çoklu duygu boyutları (örneğin, öfke, sevinç, korku gibi) gibi konular hâlâ açık araştırma alanları olarak durmaktadır. Bu tez çalışması, mevcut literatürdeki bu boşluğu doldurmayı hedeflemektedir. Öncelikle yalnızca pozitif ve negatif değil, daha ayrıntılı çoklu duygu sınıflandırmasına odaklanarak duygu çeşitliliğini ortaya koymayı amaçlamaktadır. Ayrıca, farklı sosyal medya platformlarından (örneğin Twitter, Instagram) toplanacak çeşitli veri kümeleriyle modelin genellenebilirliğini test etmekte ve dil işleme araçlarının Türkçe dilinde performansını karşılaştırmalı olarak incelemektedir. Kullanılacak algoritmaların parametre ayarları, model açıklanabilirliği ve başarı ölçütleri detaylı bir şekilde raporlanarak önceki çalışmaların eksik bıraktığı metodolojik karşılaştırma derinleştirilecektir. Sonuç olarak, bu çalışma yalnızca algoritma karşılaştırmaları değil, aynı zamanda veri çeşitliliği, duygu boyutlandırması ve Türkçe dilindeki sınırlı kaynaklara yönelik katkılarıyla literatüre önemli bir değer sağlamayı amaçlamaktadır (Tuzcu, 2020, s. 1).

Akın ve Gürsoy Şimşek, çalışmalarında sosyal medya analitiğinin işletmelere sağladığı stratejik avantajları ortaya koymuş ve sosyal medya üzerinde kullanıcıların içerik üretmesiyle ortaya çıkan büyük verinin işlenerek anlamlı bilgilere dönüştürülebileceğini göstermiştir. Özellikle, televizyon yayıncılığı örneği üzerinden gerçekleştirilen araştırmada, Twitter üzerinden toplanan yaklaşık 600.000 iletinin duygu analizi yoluyla pozitif, negatif ve nötr olarak sınıflandırıldığı belirtilmiştir. Bu sınıflandırmanın ardından elde edilen duygu verileri, reyting ölçümleriyle ilişkilendirilerek regresyon analizi yapılmıştır.

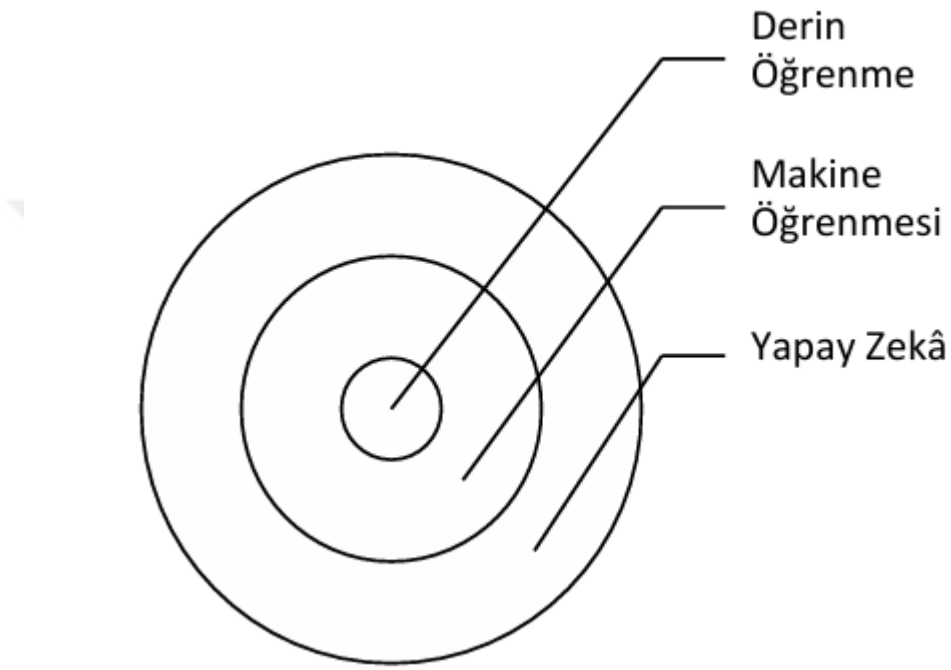
Sonuçlar, pozitif ve nötr içeriklerin reyting değerlerini artırma eğiliminde olduğunu, negatif içeriklerin ise olumsuz etki yarattığını göstermiştir. Ancak çalışmada, kullanılan duygu analizi yönteminin detayları, yöntemlerin doğruluk oranları ve literatürdeki diğer yaklaşımlar ile nasıl bir kıyaslamaya tabi tutulduğu net bir şekilde ortaya konmamıştır. Bu durum, yöntemlerin başarı düzeylerinin karşılaştırılması açısından bir belirsizlik yaratmaktadır. Ayrıca, mevcut literatür taramasında sosyal medya analitiğinin çeşitli sektörlerde kullanımına değinilmiş olsa da, çalışmanın bu alanlardaki diğer uygulamalarla olan yöntemsel veya sektörel farklarını derinlemesine analiz etmediği görülmektedir. Literatürde tespit edilen bu boşluk, özellikle Türkçe metinlerin analizinde kullanılan yöntemlerin doğruluk oranları, çok dilli veri işleme gereksinimleri ve gerçek zamanlı uygulama yetenekleri gibi konuların daha detaylı araştırılması gerektiğini göstermektedir. Bu tez çalışması, mevcut boşluğu doldurmak üzere hem Türkçe hem de çok dilli veriler üzerinde daha yüksek başarı oranları hedefleyen yeni yöntemlerin geliştirilmesini ve bu yöntemlerin performanslarının farklı veri kümeleri ve sektörler üzerinde karşılaştırmalı olarak değerlendirilmesini amaçlamaktadır. Böylece hem literatürdeki yöntem karşılaştırması eksikliği giderilecek hem de işletmeler için daha güvenilir ve gerçek zamanlı karar destek sistemleri geliştirilmesine katkı sağlanacaktır (Akın ve Gürsoy Şimşek, 2018, s. 797).

Erkırtay ve Ünal tarafından yürütülen çalışma, fikir madenciliği ve duygu analizinin toplum çevirmenliği alanında nasıl kullanılabileceğine dair yenilikçi bir yaklaşım sunmaktadır. Çalışma, sosyal medya gibi dijital mecralarda kullanıcıların ifade ettikleri duygu ve düşüncelerin, doğal dil işleme yöntemlerinden biri olan fikir madenciliği yoluyla analiz edilerek anlamlı bilgilere dönüştürülebileceğini göstermektedir. Yazarlar, bu yöntemin yalnızca pazarlama veya sağlık gibi geleneksel alanlarda değil, toplum çevirmenlerinin duygusal yükünü ölçmek ve yönetmek amacıyla da uygulanabileceğini ileri sürmektedir. Özellikle toplum çevirmenlerinin yoğun duygusal etkileşim içinde olduğu ortamlar göz önüne alındığında, bu yöntemin çevirmenlerin duygu durumlarını daha nesnel bir biçimde ortaya koyarak mesleki performanslarını iyileştirme potansiyeline sahip olduğu vurgulanmıştır. Ancak çalışmada önerilen yöntemin, kullanılan analiz düzeyleri (doküman, cümle veya özellik düzeyi) ve yöntemlerin başarı oranları gibi teknik detayları yeterince derinlemesine karşılaştırılmamıştır.

Ayrıca, önerilen modelin diğer disiplinlerdeki uygulamalarla kıyaslaması ve yöntemin doğruluğunu artırmaya yönelik teknik iyileştirmeler üzerinde durulmamıştır. Çalışma, fikir madenciliği ve duygu analizinin toplum çevirmenliğine uygulanabilirliğini tartışmakla birlikte, bu yöntemin ne ölçüde güvenilir, geçerli ve genel kullanılabilir olduğu konusunda ampirik bir doğrulama sunmamaktadır. Bu tez çalışması ise, Erkırtay ve Ünal'ın sunduğu bu yaklaşımı daha ileriye taşıyarak, farklı analiz düzeylerinde (doküman, cümle, özellik tabanlı) ve çok dilli veri kümelerinde yöntemlerin başarı oranlarını karşılaştırmayı, mevcut yöntemlerin Türkçe ve diğer diller üzerindeki performansını ölçmeyi ve özellikle gerçek zamanlı analiz kapasitesini değerlendirmeyi amaçlamaktadır. Bu sayede literatürde eksikliği hissedilen yöntem karşılaştırmaları, dilsel çeşitlilik analizi ve uygulama doğruluğu gibi boşluklar doldurularak, toplum çevirmenliği alanında daha güvenilir ve işlevsel bir duygu analizi yöntemi geliştirilmesi hedeflenmektedir. Ayrıca, bu tez çalışması, sadece çevirmenlerin değil, çeviri hizmeti alan kurumların da karar süreçlerini destekleyecek nesnel veriler üreterek sektörün genel performansını artırmaya katkı sağlamayı amaçlamaktadır (Erkırtay ve Ünal, 2021, s. 168).

Tez yazılırken kullanılan teknolojilerin tanımları ve kullanım alanları aşağıdaki gibidir:

Yapay zekâ, derin öğrenme ve makine öğrenmesi kavramları birbiriyle iç içe olduğu için sıklıkla karıştırılan kavramlardır. Şekil 1'de Derin Öğrenme, Makine Öğrenmesi ve Yapay Zekânın arasındaki ilişki gösterilmiştir. Şekil 1'de görüldüğü üzere Derin Öğrenme ve Makine Öğrenmesi, Yapay Zekânın birer dalıdır.



Şekil 1. Zekâ, Makine Öğrenmesi ve Derin Öğrenme Arasındaki İlişki

Kaynak: Dayan, A. (2022). *Doğal dil işleme ile makinelerin kendi dilini modellemesi* [Yüksek lisans tezi]. Beykent Üniversitesi.

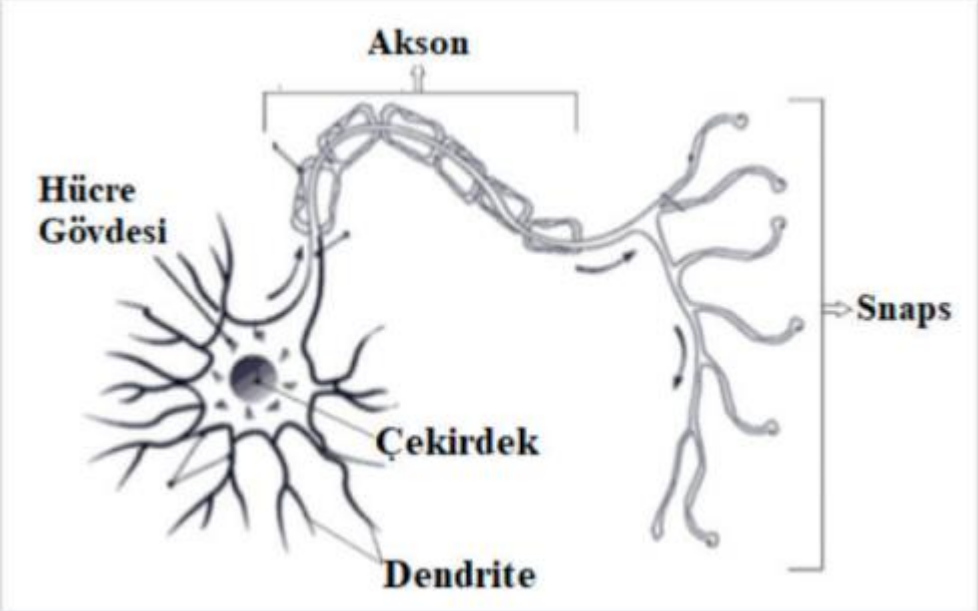
2. YAPAY ZEKA

Yapay zeka (AI), insan zekasını taklit etmeye veya insan zekasına benzer şekilde düşünebilen ve öğrenebilen sistemler geliştirmeye odaklanan bir teknoloji alanıdır. Yapay zeka, bilgisayarların problem çözme, öğrenme, mantık yürütme, karar verme ve dil anlama gibi insan zekasına özgü görevleri gerçekleştirebilmesini amaçlar.

John McCarthy, 1956'da yapay zekayı "makinelerin zeki davranış sergileme bilimi ve mühendisliği" olarak tanımlamıştır.

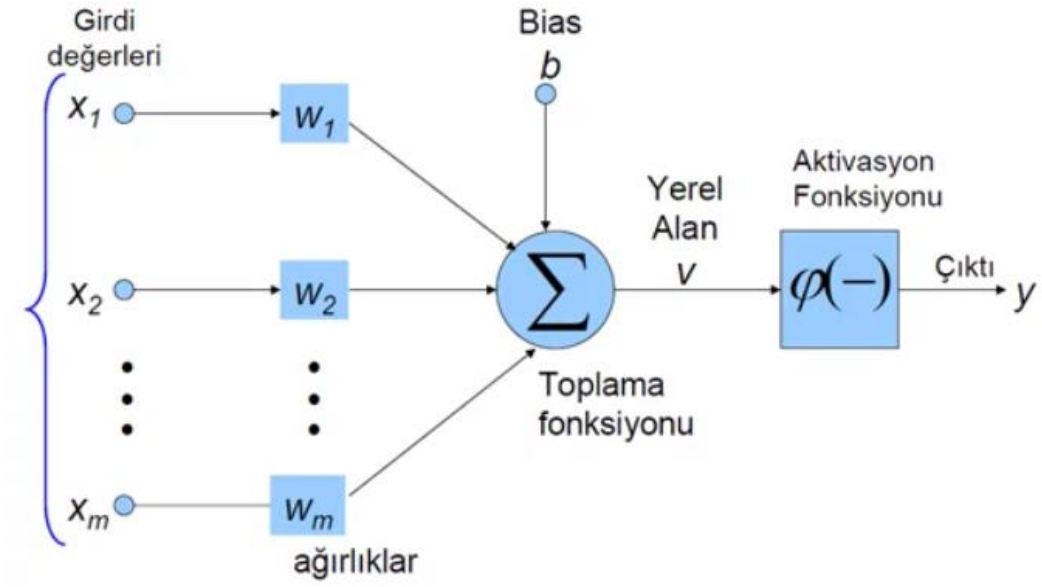
Yapay zeka, büyük miktarda veriyi analiz etmek, örüntüleri tanımak ve bu verilerden öğrenmek için algoritmalar kullanır. İşleyiş genellikle Veri Toplama ve Hazırlama, Algoritmaların Eğitimi, Tahmin ve Karar Verme şeklinde 3 aşama ile açıklanır.

Yapay sinir ağları, biyolojik sinir ağlarının çalışma prensiplerini taklit ederek geliştirilen bir yapay zeka modelidir. Temel yapı taşı, "nöron" adı verilen hesaplama birimleridir. Bu nöronlar, katmanlar halinde düzenlenerek veriyi işler ve belirli bir görevi yerine getirmek için eğitilir. Yapay sinir ağları, genellikle büyük miktarda veri ile eğitilerek, örüntü tanıma, sınıflandırma, tahmin ve karar verme gibi görevlerde kullanılır.



Şekil 2. Biyolojik Sinir Hücresi

Kaynak: Örender, A. M. ve Giray Yakut, S. (2019). Ülke kredi notlarını etkileyen faktörlerin çeşitli sınıflandırma analizleri ile incelenmesi. *Ekoist: Journal of Econometrics and Statistics*, *30*, 77–93. https://doi.org/10.26650/ekoist.2019.30.0019



Şekil 3. Yapay Sinir Hücresi

Kaynak: Bildirici, F. (2018, Temmuz 12). *Yapay zeka eğitim serisi — Bölüm 14: Nöral ağlar nasıl inşa edilir?* Medium. 10 Mayıs 2025 tarihinde https://medium.com/@fatihbildirici.dev/yapay-zeka-e%C4%9Fitim-serisi-b%C3%B6l%C3%BCm-14-696e146d3d52 adresinden edinilmiştir.

2.1. Yapay Sinir Ağlarının Temel Bileşenleri

Nöronlar(Neurons):

Yapay sinir ağlarında, her bir nöron, veri işleyen temel birimlerdir. Nöronlar, ağdaki diğer nöronlarla ağırlıklı bağlantılar aracılığıyla bilgi iletimi sağlar.

- **Katmanlar (Layers):** Yapay sinir ağları katmanlardan oluşur. Temelde üç tür katman bulunur:
- **Giriş Katmanı (Input Layer):** Modelin aldığı veriyi temsil eder.
- **Gizli Katmanlar (Hidden Layers):** Bu katmanlar, veriyi işleyip anlamlı çıkışlar üretir.
- **Çıkış Katmanı (Output Layer):** Modelin son sonucunu verir.
- **Ağırlıklar (Weights) ve Biaslar (Biases):** Ağırlıklar, nöronlar arasındaki bağlantıları kontrol eden değerlerdir ve öğrenme sırasında güncellenir. Bias ise, çıkışı değiştiren sabit bir değerdir ve ağın daha esnek öğrenmesini sağlar.

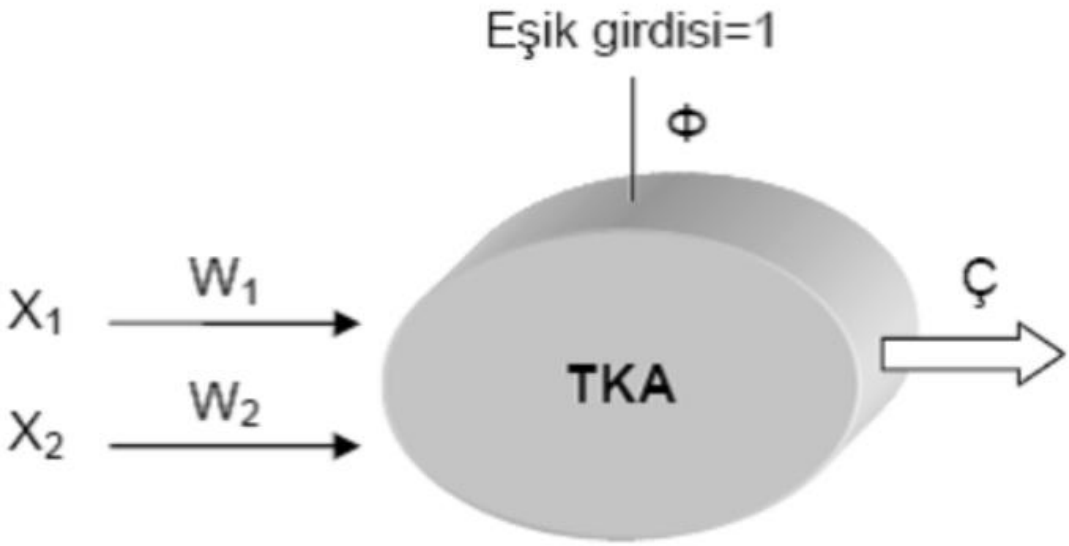
- **Aktivasyon Fonksiyonları (Activation Functions):** Aktivasyon fonksiyonları, nöronların çıktılarını belirlemek için kullanılır. Yaygın kullanılan fonksiyonlar arasında Sigmoid, ReLU (Rectified Linear Unit), ve Tanh bulunur.

2.2. Yapay Sinir Ağ Modelleri

Yapay sinir ağı modelleri tek katmanlı algılayıcılar, çok katmanlı algılayıcılar, ileri beslemeli yapay sinir ağları ve geri beslemeli yapay sinir ağları olarak dört grupta incelenebilir.

2.2.1. Tek Katmanlı Yapay Sinir Ağları

Tek katmanlı yapay sinir ağı, en basit yapıya sahip yapay sinir ağı türüdür. Bu model, yalnızca giriş katmanı ve çıkış katmanı içerir ve gizli katman bulunmaz. Bu nedenle, ağın yapısı oldukça basittir ve doğrusal olan problemlerin çözümlenmesini yapabilir.

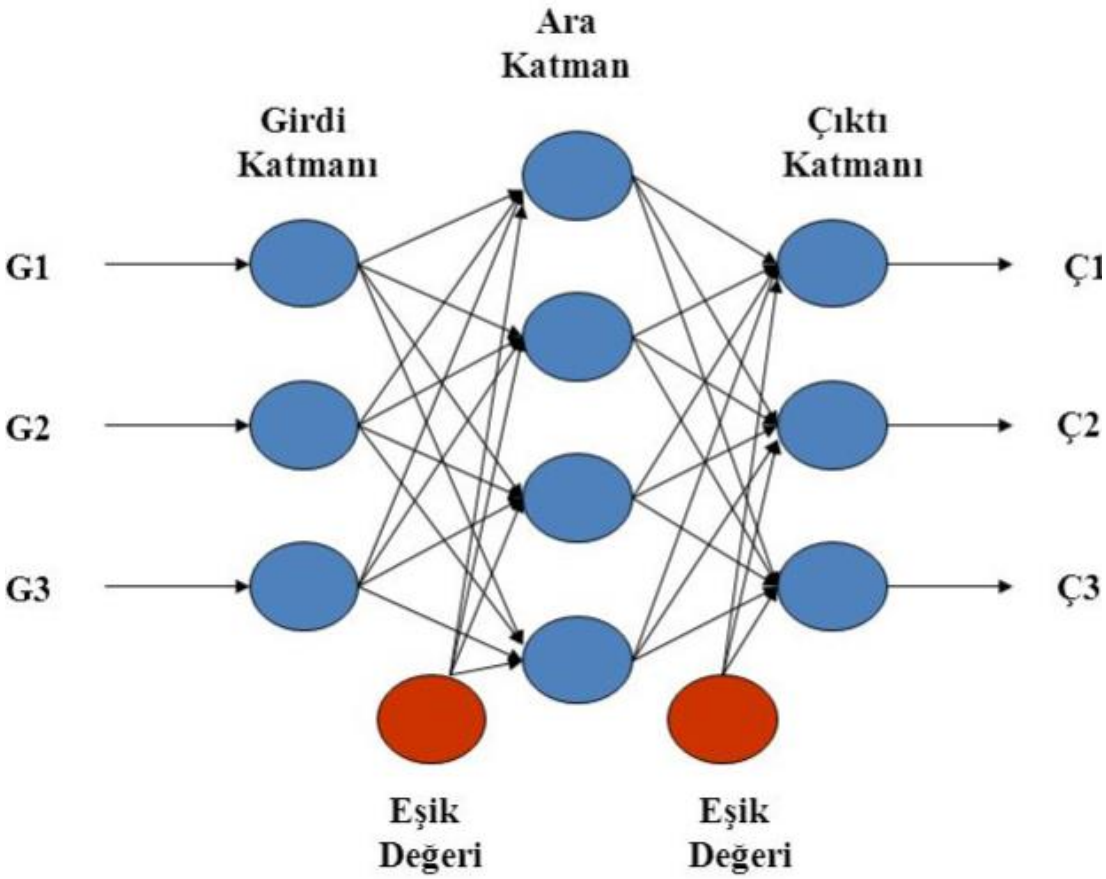


Şekil 4. Tek Katmanlı Yapay Sinir Ağı Modeli

Kaynak: Öztürk, K. ve Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, *6*(2), 25–36.

2.2.2. Çok Katmanlı Yapay Sinir Ağları

Bu yapılar, daha karmaşık problemleri çözmek için tasarlanmış, birden fazla gizli katmana sahip yapay sinir ağlarıdır. Her katman, bir önceki katmandan gelen çıkışı alıp işleyerek daha soyut ve anlamlı özellikler öğrenir. Aşağıda çok katmanlı yapay sinir ağlarının detaylı açıklamasını bulabilirsiniz.

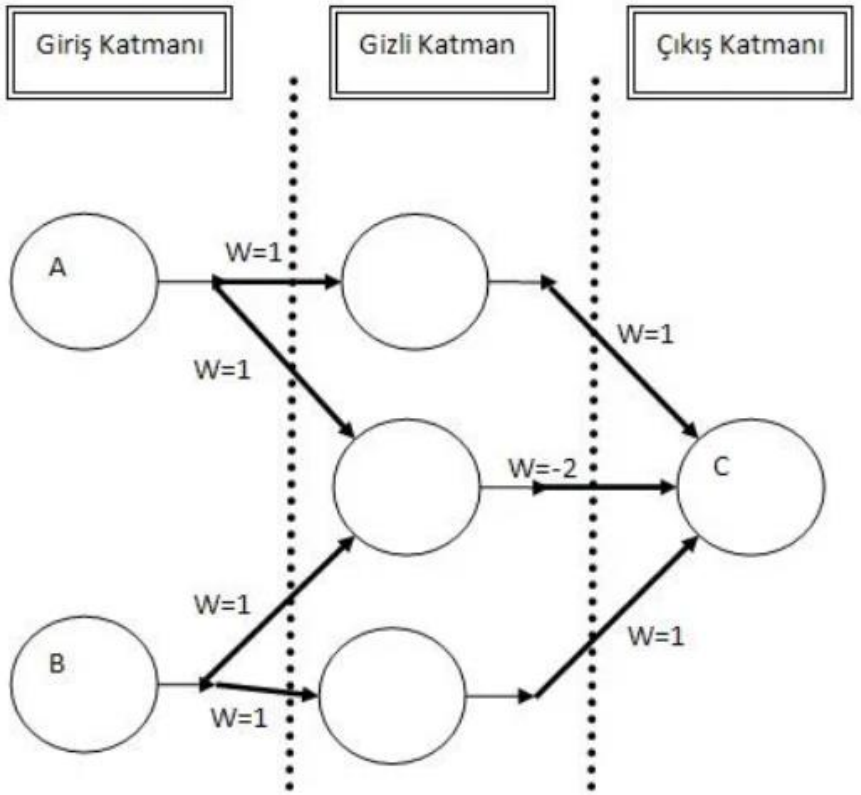


Şekil 5. Çok Katmanlı Yapay Sinir Ağı Modeli

Kaynak: Arıkanlı, T. (2022, Temmuz 5). *Yapay zeka, makine öğrenmesi ve derin öğrenme kavramları*. Medium. 5 Mayıs 2025 tarihinde https://medium.com/@tuna.45.309/yapay-zeka-makine-%C3%B6%C4%9Frenmesi-ve-derin-%C3%B6%C4%9Frenme-kavramlar%C4%B1-5b3bb67da0ae adresinden edinilmiştir.

2.2.3. İleri Beslemeli Yapay Sinir Ağları

İleri Beslemeli Yapay Sinir Ağları yapay, sinir ağlarının en temel ve yaygın türlerinden biridir. Bu ağlar, verilerin yalnızca bir yönde, yani girişten çıkışa doğru hareket ettiği ağlardır. Bu tür ağlarda veriler, her katman üzerinden bir defa geçer ve geri dönüş (feedback) veya döngüler (loops) bulunmaz. Yani, ileri beslemeli ağlar, yalnızca verinin girişten çıkışa doğru ilerlediği ve bu ilerleyişin bir yöne dayalı olduğu ağlardır.

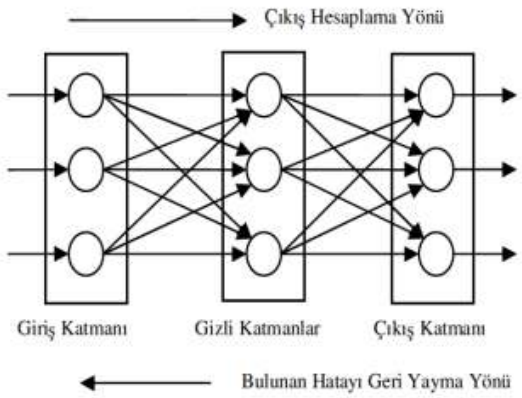


Şekil 6. İleri Beslemeli Yapay Sinir Ağı Modeli

Kaynak: Şeker, Ş. E. (2008, Ekim 5). *Yahut problemi (özel veya problemi (XOR problem, exclusive or))*. Bilgisayar Kavramları. 28 Nisan 2025 tarihinde https://bilgisayarkavramlari.com/2008/10/05/ozel-veya-problemi-xor-problem-exclusive-or/ adresinden edinilmiştir.

2.2.4. Geri Beslemeli Yapay Sinir Ağları

Geri Beslemeli Yapay Sinir Ağları, ileri beslemeli yapay sinir ağlarının aksine, veriyi yalnızca bir yönde değil, geçmiş bilgileri de dikkate alarak işler. RNN'ler, ağın her bir zaman adımında önceki çıktıları da kullanarak öğrenme sürecini daha dinamik hale getirir. Bu, özellikle zaman serisi verileri, dil modeli veya sıralı veri analizi gibi durumlarda oldukça faydalıdır.



Şekil 7. Geri Beslemeli Yapay Sinir Ağı Modeli

Kaynak: Öztürk, K. ve Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, *6*(2), 25–36.

3. MAKİNE ÖĞRENMESİ

Makine öğrenmesi (ML), bilgisayarların insan müdahalesi olmadan veri üzerinden öğrenme yeteneğini geliştiren bir yapay zeka dalıdır. Geleneksel programlamada, bilgisayarlar belirli kurallara göre işlem yaparken, makine öğrenmesinde bilgisayarlar büyük miktarda veri ve örüntüler üzerinden kendiliğinden kararlar almayı öğrenir. Bu süreç, belirli algoritmalar ve modeller kullanılarak gerçekleştirilir. Makine öğrenmesinde, iki ana problem tipi bulunmaktadır: Kümeleme ve sınıflandırma problem tipleri.

Kümeleme: Verilerin benzerlikleri yakalanarak, yakısanama yapılarak, birbirleriyle ilişkilerini ortaya koyarak bir gruplama mantığı taşıyan problem tipine kümeleme problemi denir. Etiket bilgisi mevcut değildir. Veri setinin ouput'ları ile ilgili bilgi mevcut değildir. Elde bulunan verilerin sadece nitelikleri ve özellikleri mevcuttur.

Sınıflandırma: Veri setinin input ve output bilgileri mevcuttur. Etiket bilgisinin bilindiği problem tipine sınıflandırma problemi denir. Bu tip problemlerde input ve output değerleri bilindiğinden dolayı tahminleme yapılacak input değerleri için output değerleri hakkında kestirim yapılabilir. Makine öğrenmesi, dört ana kategoride incelenir: denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve takviyeli öğrenme**.**

3.1. Denetimli Öğrenme

Denetimli öğrenmede, modele öğrenmesi için etiketli veri sağlanır. Modelin giriş (input) ve çıkış (output) değerleri bilinir. Verilen giriş bilgilerine karşılık hangi çıkışların elde edildiği analiz edilir ve bu ilişki üzerinden tahminleme gerçekleştirilir. Bu sayede, output'u bilinmeyen yeni input değerleri hakkında kestirim yapılabilir.

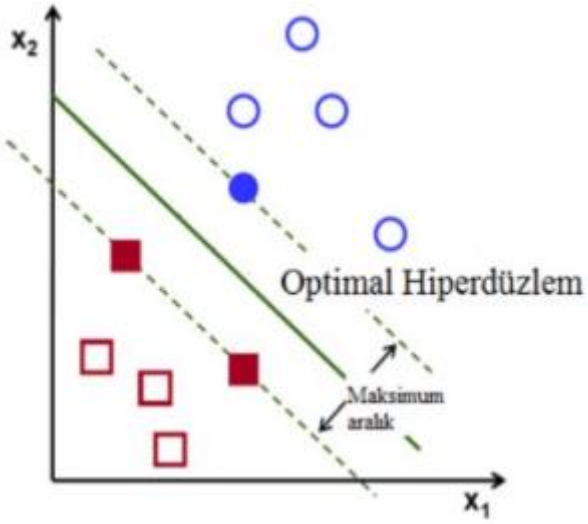
Denetimli öğrenme genellikle sınıflandırma ve regresyon problemlerinde kullanılır. Sınıflandırma problemlerinde, veri belirli kategorilere ayrılırken; regresyon problemlerinde, sürekli değerlerin tahmini yapılır. Örnek Algoritmalar:

- Lineer Regresyon**:** Bir bağımlı değişken ile bir veya birden fazla bağımsız değişken arasındaki ilişkiyi modellemek için kullanılır.
- Lojistik Regresyon**:** Verileri iki sınıfa ayırmak için kullanılan bir sınıflandırma algoritmasıdır.
- Destek Vektör Makineleri (SVM)**:** Verileri sınıflandırmak için en uygun ayrım sınırını (hyperplane) bulan bir algoritmadır.
- Naive Bayes**:** Olasılık teorisine dayalı bir sınıflandırma algoritmasıdır. Bayes Teoremi'ni kullanarak, verilen özelliklerin belirli bir sınıfa ait olma olasılığını hesaplar. Özellikle metin sınıflandırma, spam tespiti ve duygu analizi gibi uygulamalarda yaygın olarak kullanılır. Naive Bayes'in temel varsayımı, özelliklerin birbirinden bağımsız olduğu yönündedir.

3.1.1. Destek Vektör Makineleri (SVM)

Destek Vektör Makinesi (SVM) algoritması, N-boyutlu bir uzayda (N, özellik sayısı) veri noktalarını sınıflandırabilen bir hiperdüzlem bulmayı amaçlar. İki sınıf arasındaki veriyi ayırabilmek için farklı hiperdüzlemler seçilebilir. Ancak, amacımız, her iki sınıf arasındaki veri noktaları arasındaki mesafeyi en üst düzeye çıkaran, yani maksimum marj (sınır) olan düzlemi bulmaktır. Marj mesafesinin büyütülmesi, gelecekteki veri noktalarının doğru şekilde sınıflandırılabilmesi için daha güvenli bir temel sağlar. Hiperdüzlemler, veri noktalarını sınıflandırmada kullanılan karar sınırlarıdır ve her iki tarafında yer alan veri noktaları farklı sınıflarla ilişkilendirilebilir. Ayrıca, hiperdüzlemin boyutu, giriş özelliklerinin sayısına bağlıdır.

Eğer özellik sayısı 2 ise, hiperdüzlem bir çizgi olarak görünür. Özellik sayısı 3 ise, hiperdüzlem bir düzleme dönüşür. Özellik sayısı 3'ü geçtiğinde ise, geometrik olarak anlamak ve görselleştirmek oldukça zorlaşır.



Şekil 8. SVM Destek Vektörleri ve Hiper Düzlemi

Kaynak: Bayrakdar, S., Akgün, D. ve Yücedağ, İ. (2016). Yüz ifadelerinin otomatik analizi üzerine bir literatür çalışması. *Sakarya Üniversitesi Fen Bilimleri Dergisi*, *20*(2), 383–398.

Şekil 8' de iki farklı sınıfa ait örnekler bulunmaktadır. Destek vektörleri, ayırıcı düzleme en yakın olan ve bu düzlemin konumunu ve yönünü belirleyen veri noktalarıdır. Bu destek vektörlerini kullanarak sınıflandırıcının marjını en yüksek seviyeye çıkarmaya çalışırız. Destek vektörlerinin kaldırılması, ayırıcı düzlemin konumunun değişmesine yol açar. Bu noktalar, destek vektör makinesini oluştururken önemli bir rol oynar ve sınıflandırma işlemine yön verir.

3.1.2. Naive Bayes

Naive Bayes sınıflandırıcısı, adını 18. yüzyılda olasılık teorisine önemli katkılarda bulunan İngiliz matematikçi Thomas Bayes'ten almıştır. Naive Bayes sınıflandırıcısı, temel olarak Bayes Teoremine dayanan ve olasılıksal hesaplamalarla çalışan bir makine öğrenme yöntemidir. Naive Bayes algoritması, her özelliğin sınıfa bağımsız olarak katkıda bulunduğunu varsayarak çalışır. Bu "naive" yani saf varsayım, gerçek dünyadaki verilerde her zaman tam olarak geçerli olmasa da, sınıflandırıcıyı oldukça hızlı ve verimli hale getirmektedir.

Özellikle boyutları büyük olan veri setleri üzerinde çalışırken, Naive Bayes yöntemi ile parametre tahminlerini yapmak oldukça kolaydır. Bu yöntem, basit bir yapıya sahip olmasına rağmen, bazı durumlarda çok daha karmaşık ve hesaplama açısından maliyetli olan diğer sınıflandırma algoritmalarına kıyasla yüksek başarı oranları sunmaktadır. Çeşitli araştırmalar ve uygulamalar, Naive Bayes sınıflandırıcısının metin madenciliği, spam filtreleme, duygu analizi ve tıbbi teşhis gibi birçok farklı alanda etkili sonuçlar verdiğini göstermiştir.

Bayes Teoremi, olasılık teorisinde rassal değişkenler arasındaki koşullu ve marjinal olasılıklar arasındaki ilişkiyi tanımlayan temel bir prensiptir. Bu teorem sayesinde, bir olayın belirli koşullar altında gerçekleşme olasılığı hesaplanabilir. Naive Bayes sınıflandırıcısı da bu teoremi temel alarak, bir verinin belirli bir sınıfa ait olma olasılığını hesaplar ve en yüksek olasılığa sahip sınıfı tahmin sonucu olarak belirler.

$$P(A\backslash B) = \frac{P(B\backslash A)P(A)}{P(B)}$$

Şekil 9. Naive Bayes Teoremi Formülü

Kaynak: Bıçakcı, K. (2021, June 12). *Olaylara bayesci çıkarım ile bakmak | bayesci vs frekansçı istatistik*. Medium. 5 Mayıs 2025 tarihinde https://frightera.medium.com/olaylara-bayesci-%C3%A7%C4%B1kar%C4%B1m-ile-bakmak-bayesci-vs-frekans%C3%A7%C4%B1-i%CC%87statistik-b59d3a9f5466 adresinden edinilmiştir.

P(A\B): B olayı gerçekleşirse A olayının meydana gelme olasılığı

P(B\A): A olayı gerçekleşirse B olayının meydana gelme olasılığı

P(A): A olayının ön olasılığı

P(B): B olayının ön olasılığı

Naive Bayes sınıflandırma modeli, bir dizi bağımsız değişken (özellik) ile bunlara bağlı bir sonuç değişkeninden oluşan bir olasılık tabanlı sınıflandırma yöntemidir. Bu modelde hesaplamalar, her bir sınıf için ayrı ayrı yapılır ve verilen bir veri noktasının belirli bir sınıfa ait olma olasılığı hesaplanarak, en yüksek olasılığa sahip sınıf nihai sınıf olarak belirlenir.

Bayes Teoreminden yola çıkarak, Naive Bayes sınıflandırma sürecini daha iyi anlamak için elimizde n adet farklı sınıf olduğunu varsayalım. Elimizdeki sınıfları S harfi ile, herhangi bir sınıfa ait olup olmadığı bilinmeyen bir veri örneğini ise X harfi ile ifade edelim. Bu durumda, X verisinin hangi sınıfa ait olduğu, Naive Bayes algoritması kullanılarak hesaplanabilir. Model, X'in her bir sınıfa ait olma olasılığını değerlendirerek en yüksek olasılığa sahip sınıfa atanmasını sağlar. Böylece, X en uygun **S** sınıfına yerleştirilmiş olur.

Burada önemli olan nokta, X'in hangi sınıfa atanacağına dair kararın tamamen olasılıksal bir temele dayanmasıdır. Naive Bayes sınıflandırıcı, X'in hangi sınıfa ait olabileceğini belirlerken belirli bir olasılık değeri hesaplar. Bu süreçte, X'i tanımlayan özellikler m boyutlu bir vektör olarak ele alınır ve model, bu özelliklerin birbirinden bağımsız olduğunu varsayar. Yani, bir özelliğin değeri, başka bir özelliğin değeri hakkında herhangi bir bilgi içermez ve tüm özellikler eşit derecede önemli kabul edilir. Bu "bağımsızlık" varsayımı, Naive Bayes sınıflandırıcısının hızlı ve etkili bir şekilde çalışmasını sağlayan temel prensiplerden biridir.

Eğer tüm sınıflar için P(X) değeri sabit kabul edilirse, bir verinin Si sınıfına ait olma olasılığı P(X|Si) * P(Si) şeklinde hesaplanır. Burada P(Si)**,** Si sınıfının genel olasılığını temsil eder. Bu olasılık, sınıfa ait eğitim verilerinin toplam eğitim verilerine oranı olarak belirlenir. Eğer Si sınıfına ait eğitim örneklerinin sayısını Oi, tüm eğitim örneklerinin toplam sayısını ise O olarak ifade edersek, P(Si) değeri hesaplanır.

Eğer sınıfların ön olasılıkları bilinmiyorsa, genellikle tüm sınıfların eşit olasılığa sahip olduğu varsayılır. Bu durumda P(S1) = P(S2) = ... = P(Sn) kabul edilir. Böyle bir durumda, P(X|Si) ifadesi, X verisinin Si sınıfına ait olma olasılığını belirlemek için kullanılır.

Ancak, sınıfların ön olasılıkları biliniyorsa, en doğru hesaplama P(X|Si) ***** P(Si) formülüyle yapılır. Bilinmeyen bir X örneğinin hangi sınıfa ait olduğunu belirlemek için, her bir Si sınıfı için P(X|Si) * P(Si) değeri hesaplanır ve X değeri, en yüksek olasılığa sahip Si sınıfına atanır.

3.1.3. Rastgele Orman (Random Forest)

Rastgele Orman (Random Forest), hem regresyon hem de sınıflandırma problemleri için kullanılabilen güçlü ve çok yönlü bir topluluk öğrenme (ensemble learning) algoritmasıdır. Birden çok karar ağacının (decision tree) sonuçlarını birleştirerek tek bir daha kararlı ve doğru tahmin elde etmeyi amaçlar. Leo Breiman tarafından geliştirilen Rastgele Orman, karar ağaçlarının basitliğini ve anlaşılırlığını, topluluk öğrenmesinin gücüyle birleştirerek yüksek performans ve genelleme yeteneği sunar. Rastgele Ormanın Temel Mantığı, Rastgele Orman'ın temelinde iki önemli kavram yatar.

Rastgele Örnekleme (Bootstrap Aggregating - Bagging): Eğitim veri setinden rastgele ve tekrarlı örnekler (bootstrap örnekleri) alınarak birden çok farklı alt veri seti oluşturulur. Her bir karar ağacı, bu farklı alt veri setleri üzerinde eğitilir. Bu rastgele örnekleme, her bir ağacın farklı bir perspektife sahip olmasını sağlar ve modelin eğitim verisine aşırı uyum (overfitting) riskini azaltır.

Rastgele Özellik Seçimi: Her bir karar ağacının düğümünde (node) en iyi bölmeyi (split) belirlerken, tüm özellikler yerine rastgele seçilmiş bir alt küme dikkate alınır. Bu rastgele özellik seçimi, ağaçlar arasındaki korelasyonu azaltır ve modelin genelleme yeteneğini daha da artırır. Farklı ağaçlar, veri setindeki farklı özelliklerin önemini vurgulayarak daha çeşitli tahminler yaparlar.

Rastgele Ormanın Çalışma Adımları: Rastgele Orman'ın 2 önemli çalışma adımı şunlardır.

Bootstrap Örneklemesi: Orijinal eğitim veri setinden rastgele ve tekrarlı olacak şekilde N sayıda örneklem (aynı boyutta) alınır, burada N oluşturulacak karar ağacı sayısını temsil eder. Bir veri noktası aynı örneklemde birden fazla kez yer alabilir. Karar Ağacı Eğitimi: Her bir bootstrap örneklemi için bir karar ağacı eğitilir. Ancak, her düğümde en iyi bölmeyi seçerken, tüm özellikler yerine rastgele seçilmiş m sayıda özellik (tipik olarak toplam özellik sayısının karekökü) arasından en iyi bölmeyi sağlayan özellik ve eşik değeri belirlenir. Bu rastgele özellik seçimi, ağaçlar arasındaki çeşitliliği artırır. Tahmin yapma 2 türlüdür:

- Sınıflandırma: Yeni bir girdi için, ormandaki her bir karar ağacı bir sınıf tahmini yapar. Rastgele Orman'ın nihai tahmini, ağaçların yaptığı tahminlerin çoğunluğu (oy birliği - majority vote) ile belirlenir.
- Regresyon: Yeni bir girdi için, ormandaki her bir karar ağacı bir sayısal değer tahmini yapar. Rastgele Orman'ın nihai tahmini, ağaçların yaptığı tahminlerin ortalaması alınarak belirlenir.

Rastgele Ormanın Avantajları:

- Yüksek Doğruluk: Birden çok karar ağacının sonuçlarını birleştirerek genellikle tek bir karar ağacından daha yüksek doğruluk elde eder.
- Aşırı Uyum Riskini Azaltma: Rastgele örnekleme ve rastgele özellik seçimi sayesinde aşırı uyum (overfitting) riski önemli ölçüde azalır.
- Özellik Önemini Belirleme: Eğitim sürecinde hangi özelliklerin tahminler üzerinde daha etkili olduğunu belirlemek için bir mekanizma sunar (feature importance). Bu, veri setindeki en önemli değişkenleri anlamak için faydalıdır.
- Eksik Verilere Karşı Dayanıklılık: Bazı durumlarda eksik verilere karşı diğer algoritmalara göre daha dayanıklı olabilir.
- Ayarlama Gereksinimi Nispeten Az: Diğer bazı makine öğrenmesi algoritmalarına kıyasla hiperparametre ayarlamasına daha az duyarlıdır. İyi varsayılan parametrelerle bile genellikle iyi performans gösterir.
- Büyük Veri Setleriyle İyi Çalışma Potansiyeli: Paralelleştirilebilir yapısı sayesinde büyük veri setleriyle etkili bir şekilde çalışabilir. Her bir karar ağacı bağımsız olarak eğitilebilir.
- Hem Sınıflandırma Hem de Regresyon İçin Kullanılabilir: Çok yönlü bir algoritmadır ve farklı türdeki tahmin problemlerine uygulanabilir.
- Out-of-Bag (OOB) Hata Tahmini: Bootstrap örneklemesi sırasında her bir ağacın eğitiminde kullanılmayan veri noktaları (out-of-bag örnekleri) modelin genelleme performansını tahmin etmek için kullanılabilir. Bu, ayrı bir doğrulama setine olan ihtiyacı azaltabilir.

Rastgele Ormanın Dezavantajları:

- Yorumlanabilirlik Kaybı: Tek bir karar ağacına kıyasla, yüzlerce veya binlerce ağacın birleşimiyle oluşan Rastgele Orman modelinin yorumlanması daha zordur. Hangi özelliklerin nihai tahmini nasıl etkilediğini anlamak karmaşıklaşabilir.
- Eğitim Süresi: Çok sayıda karar ağacının eğitilmesi gerektiği için, tek bir karar ağacına veya bazı diğer algoritmalara kıyasla eğitim süresi daha uzun olabilir.
- Bellek Tüketimi: Çok sayıda karar ağacını saklamak önemli miktarda bellek gerektirebilir.
- Gerçek Zamanlı Tahminlerde Gecikme: Çok sayıda ağacın tahminlerini birleştirmek, tek bir modelin tahminine göre biraz daha zaman alabilir. Bu, düşük gecikme gerektiren gerçek zamanlı uygulamalarda bir dezavantaj olabilir.

3.1.4. Özellik Seçimi (Feature Selection)

Özellik seçimi (bazen değişken seçimi, öznitelik seçimi olarak da adlandırılır), makine öğrenmesi ve istatistiksel modelleme süreçlerinde kullanılan kritik bir adımdır. Temel amacı, bir veri setindeki tüm olası girdi özelliklerinden (değişkenlerden) modelin performansı üzerinde en büyük etkiye sahip olan, en ilgili ve bilgilendirici alt kümeyi belirlemektir. Başka bir deyişle, amaç, modelin öğrenme sürecine katkıda bulunmayan, gereksiz, yanıltıcı veya birbiriyle yüksek oranda ilişkili (redundant) özellikleri eleyerek daha sade, daha anlaşılır ve daha iyi performans gösteren modeller oluşturmaktır.

Özellik seçimi, veri ön işleme aşamasının önemli bir parçasıdır ve modelin başarısını doğrudan etkileyebilir. Yüksek boyutlu veri setlerinin yaygınlaştığı günümüzde, modelin eğitileceği özellik sayısının azaltılması, hem hesaplama maliyetlerini düşürmekte hem de modelin genelleme yeteneğini artırmada hayati bir rol oynamaktadır.

Özellik Seçiminin Temel Amaçları:

- Model Performansını İyileştirme: En ilgili özelliklerin seçilmesi, modelin tahmin doğruluğunu ve kararlılığını artırabilir. Gereksiz özellikler, modelin gürültüye odaklanmasına (aşırı uyum - overfitting) ve dolayısıyla yeni veriler üzerindeki performansının düşmesine neden olabilir.
- Hesaplama Maliyetlerini Azaltma: Daha az sayıda özellikle eğitilen modeller, daha hızlı eğitilir ve daha az bellek kaynağı tüketir. Bu, özellikle büyük veri setleriyle çalışırken önemli bir avantaj sağlar.
- Modelin Yorumlanabilirliğini Artırma: Daha az sayıda özellik içeren modellerin anlaşılması ve yorumlanması daha kolaydır. Hangi özelliklerin modelin tahminlerinde en etkili olduğunu belirlemek, problem alanı hakkında değerli içgörüler sunabilir.
- Aşırı Uyumu (Overfitting) Önleme: Gereksiz özelliklerin elenmesi, modelin eğitim verisine aşırı derecede uyum sağlamasını ve yeni verilere karşı zayıf genelleme yapmasını engellemeye yardımcı olur.
- Veri Anlayışını Derinleştirme: Özellik seçimi süreci, hangi özelliklerin hedef değişkenle daha güçlü bir ilişkiye sahip olduğunu ortaya çıkararak veri setinin yapısı ve değişkenler arasındaki ilişkiler hakkında daha iyi bir anlayış sağlar.

Özellik Seçimi Yöntemleri:

Özellik seçimi yöntemleri genel olarak üç ana kategoriye ayrılabilir:

Filtre Yöntemleri (Filter Methods): Bu yöntemler, özellikleri hedef değişkenle olan ilişkilerini istatistiksel metrikler (örneğin korelasyon, bilgi kazancı, ki-kare testi) kullanarak değerlendirir ve bir skor atar. Özellikler bu skorlara göre sıralanır ve belirli bir eşik değerine göre en iyi özellikler seçilir. Bu yöntemler modelden bağımsızdır ve genellikle hesaplama açısından hızlıdır. Örnekler:

- Korelasyon Katsayısı**:** Sürekli hedef değişkenler için özelliklerle hedef arasındaki doğrusal ilişkiyi ölçer. Yüksek korelasyona sahip özellikler seçilebilir.
- Ki-Kare Testi (Chi-Squared Test): Kategorik hedef değişkenler için kategorik özelliklerle hedef arasındaki bağımsızlığı test eder. Hedefle en bağımlı olan özellikler seçilebilir.

- Bilgi Kazancı (Information Gain) / Karar Ağacı Tabanlı Önem Skorları**:** Özellikle karar ağacı ve türevlerinde (örneğin Rastgele Orman, Gradyan Artırma) bir özelliğin hedef değişkeni ne kadar iyi ayırdığını ölçer.
- Varyans Eşiği (Variance Threshold): Düşük varyansa sahip özellikleri (neredeyse sabit değerler) eleyerek bilgi taşımayan değişkenleri temizler.
- ANOVA F-testi: Sürekli özelliklerin kategorik hedef değişken üzerindeki etkisini istatistiksel olarak değerlendirir.

Sarıcı Yöntemler (Wrapper Methods): Bu yöntemler, farklı özellik alt kümelerini kullanarak bir makine öğrenmesi modelini eğitir ve modelin performansına göre en iyi özellik alt kümesini seçmeye çalışır. Bu yöntemler, modelin performansını doğrudan optimize etmeyi amaçladıkları için genellikle daha iyi sonuçlar verebilirler, ancak hesaplama açısından daha maliyetlidirler çünkü birçok farklı model eğitimi gerektirirler. Örnekler:

- İleriye Doğru Seçim (Forward Selection): Boş bir küme ile başlanır ve her adımda modele performansı en çok artıran özellik eklenir. Bu işlem, belirli bir sayıda özellik seçilene veya performans iyileşmesi durana kadar devam eder.
- Geriye Doğru Eleme (Backward Elimination): Tüm özelliklerle başlanır ve her adımda modelin performansını en az etkileyen özellik çıkarılır. Bu işlem, belirli bir sayıda özellik kalana veya performans düşüşü kabul edilemez hale gelene kadar devam eder.
- Özyinelemeli Özellik Eleme (Recursive Feature Elimination - RFE): Bir model (örneğin SVM, lojistik regresyon) tekrar tekrar eğitilir ve her adımda en az önemli olduğu tahmin edilen özellikler elenir. Kalan özellikler, modelin performansına göre sıralanır.
- En İyi Alt Küme Seçimi (Best Subset Selection): Tüm olası özellik alt kümeleri denenir ve en iyi performansı veren alt küme seçilir. Ancak, özellik sayısı arttıkça bu yöntem hesaplama açısından çok maliyetli hale gelir.

Gömülü Yöntemler (Embedded Methods)**:** Bu yöntemler, özellik seçimini modelin eğitim sürecine entegre eder. Modelin kendisi, hangi özelliklerin daha önemli olduğunu belirleyen mekanizmalar içerir. Bu yöntemler, filtre yöntemlerinin hesaplama verimliliği ile sarıcı yöntemlerinin model odaklılığını birleştirmeyi amaçlar. Örnekler:

- L1 Düzenlileştirme (Lasso): Doğrusal modellerin (örneğin doğrusal regresyon, lojistik regresyon) eğitiminde kullanılan bir tekniktir. L1 normu, bazı özelliklerin katsayılarını sıfıra düşürerek etkili bir özellik seçimi yapar.
- Ağaç Tabanlı Modellerin Özellik Önem Skorları (Feature Importance from Tree-based Models): Karar ağaçları, rastgele ormanlar ve gradyan artırma gibi algoritmalar, her bir özelliğin modelin tahminlerinde ne kadar önemli olduğuna dair bir skor üretir. Bu skorlar, özellik seçimi için kullanılabilir.

Özellik Seçimi Sürecinde Dikkat Edilmesi Gerekenler:

- Veri Ön İşleme: Özellik seçimi uygulamadan önce verinin temizlenmesi, eksik değerlerin giderilmesi ve gerekirse ölçeklendirilmesi önemlidir.
- Amaç ve Bağlam: Hangi özelliklerin "en iyi" olduğu, problemin amacına ve bağlamına bağlıdır. Örneğin, tahmin doğruluğu öncelikli ise sarıcı yöntemler daha uygun olabilirken, yorumlanabilirlik önemliyse filtre veya gömülü yöntemler tercih edilebilir.
- Model Seçimi: Bazı özellik seçimi yöntemleri belirli model türleriyle daha iyi çalışır. Örneğin, L1 düzenlileştirme doğrusal modeller için etkilidir.
- Doğrulama (Validation): Seçilen özellik alt kümesinin model üzerindeki gerçek performansını değerlendirmek için çapraz doğrulama (cross-validation) gibi güvenilir değerlendirme yöntemleri kullanılmalıdır.
- Özellik Etkileşimleri: Basit özellik seçimi yöntemleri, özellikler arasındaki etkileşimleri göz ardı edebilir. Bu durumda, özellik mühendisliği teknikleriyle etkileşim terimleri oluşturmak faydalı olabilir.

3.1.4.1. Ki-Kare Testi (Chi-Squared Test)

Ki-Kare Testi ($\chi 2$), istatistikte kategorik (nitel) değişkenler arasındaki ilişkileri veya gözlemlenen frekansların beklenen frekanslardan anlamlı ölçüde farklı olup olmadığını değerlendirmek için kullanılan güçlü ve yaygın bir hipotez testidir. Karl Pearson tarafından geliştirilen bu test, özellikle sayılabilir verilerle çalışırken, teorik beklentilerle gerçek gözlemler arasındaki uyumu istatistiksel olarak incelemek için kullanılır. Ki-Kare testi, tek bir yöntem olmaktan ziyade, benzer prensiplere dayanan bir dizi istatistiksel testi ifade eder. En yaygın kullanılan Ki-Kare testleri şunlardır:

Ki-Kare Uyum İyiliği Testi (Chi-Squared Goodness-of-Fit Test): Bir örneklemden elde edilen frekans dağılımının, belirli bir teorik veya beklenen dağılıma uyup uymadığını test etmek için kullanılır. Amaç, gözlemlenen frekansların rastlantısal sapmalar mı yoksa teorik dağılımdan anlamlı bir farklılık mı gösterdiğini belirlemektir.

Ki-Kare Bağımsızlık Testi (Chi-Squared Test of Independence): İki veya daha fazla kategorik değişken arasında istatistiksel olarak anlamlı bir ilişki (bağımlılık) olup olmadığını test etmek için kullanılır. Amaç, bir değişkendeki kategorilerin diğer değişkendeki kategorilerin dağılımını etkileyip etkilemediğini anlamaktır.

Ki-Kare Homojenlik Testi (Chi-Squared Test of Homogeneity): İki veya daha fazla farklı popülasyondan alınan örneklemlerin belirli bir kategorik değişken açısından benzer (homojen) olup olmadığını test etmek için kullanılır.

Amaç, farklı grupların belirli bir özellik açısından aynı dağılıma sahip olup olmadığını belirlemektir.

Ki-Kare Testinin Temel Mantığı:

Ki-Kare testlerinin temelinde, gözlemlenen frekanslar (gerçekte elde edilen sayımlar) ile beklenen frekanslar (eğer değişkenler arasında bir ilişki yoksa veya teorik dağılım doğruysa beklenen sayımlar) arasındaki farkın büyüklüğünü değerlendirmek yatar. Eğer bu fark yeterince büyükse, gözlemlenen sonuçların rastlantısal olmadığı ve dolayısıyla hipotezin (örneğin bağımsızlık hipotezi) reddedilmesi gerektiği sonucuna varılır.

Ki-Kare İstatistiğinin Hesaplanması:

Ki-Kare istatistiği (χ2), aşağıdaki formül kullanılarak hesaplanır:

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Şekil 10. Ki-Kare Testi (Chi-Squared) Yöntemi Formülü

Kaynak: Yazar tarafından oluşturulmuştur.

Burada:

- Oi: i-inci kategorideki gözlemlenen frekansı temsil eder.
- Ei: i-inci kategorideki beklenen frekansı temsil eder.
- k: Toplam kategori sayısını temsil eder.

Formül, her bir kategori için gözlemlenen ve beklenen frekanslar arasındaki farkın karesinin beklenen frekansa bölünmesiyle elde edilen değerlerin toplamını alır. Büyük bir Ki-Kare değeri, gözlemlenen ve beklenen frekanslar arasında büyük bir fark olduğunu gösterir.

Beklenen Frekansların Hesaplanması:

- Uyum İyiliği Testi'nde: Beklenen frekanslar genellikle teorik bir dağılıma veya önceden belirlenmiş oranlara göre hesaplanır. Örneğin, adil bir zarda her bir yüzün eşit olasılıkta (1/6) gelmesi beklenir.
- Bağımsızlık ve Homojenlik Testlerinde: Beklenen frekanslar, eğer değişkenler arasında bağımsızlık varsa veya popülasyonlar homojen ise her bir hücrede beklenen sayıyı temsil eder. Genellikle, satır toplamı ile sütun toplamının çarpımının toplam gözlem sayısına bölünmesiyle hesaplanır:

$$E_{ij} = \frac{(\text{Satır } i \text{ Toplamı}) \times (\text{Sütun } j \text{ Toplamı})}{\text{Toplam Gözlem Sayısı}}$$

Şekil 11. Bağımsızlık ve Homojenlik Formülü

Kaynak: Yazar tarafından oluşturulmuştur.

Hipotez Testi ve Anlamlılık Değeri (p-değeri):

Hesaplanan Ki-Kare istatistiği, belirli bir serbestlik derecesi (degree of freedom - df) ile Ki-Kare dağılımıyla karşılaştırılır. Serbestlik derecesi, testin türüne göre farklılık gösterir:

- Uyum İyiliği Testi'nde: df=(Kategori Sayısı−1)
- Bağımsızlık ve Homojenlik Testlerinde:
  df=(Satır Sayısı−1)×(Sutun Sayısı−1)

Karşılaştırma sonucunda bir anlamlılık değeri (p-değeri) elde edilir. p-değeri, eğer boş hipotez (örneğin, dağılım beklenen dağılıma uyuyor veya değişkenler bağımsız) doğruysa, gözlemlenen verilere eşit veya daha uç bir sonuç elde etme olasılığını gösterir. Genellikle, önceden belirlenmiş bir anlamlılık düzeyi (α, yaygın olarak 0.05 kullanılır) ile p-değeri karşılaştırılır:

- Eğer $p \leq \alpha$ ise, boş hipotez reddedilir. Bu, gözlemlenen verilerin boş hipotez ile uyumlu olmadığı ve alternatif hipotezin (örneğin, dağılım farklı veya değişkenler bağımlı) desteklendiği anlamına gelir.
- Eğer $p > \alpha$ ise, boş hipotez reddedilemez. Bu, gözlemlenen verilerin boş hipotez ile tutarlı olduğu anlamına gelir, ancak boş hipotezin doğru olduğu kesin olarak kanıtlanamaz.

Ki-Kare Testinin Kullanım Alanları:

Ki-Kare testleri, kategorik verilerin olduğu birçok alanda yaygın olarak kullanılır:

- Pazar Araştırması: Müşteri tercihleri ile demografik özellikler arasındaki ilişkileri analiz etmek.
- Sosyal Bilimler: Farklı grupların tutumları veya davranışları arasındaki farklılıkları incelemek.
- Sağlık Bilimleri: Risk faktörleri ile hastalıkların görülme sıklığı arasındaki ilişkileri değerlendirmek.
- Eğitim: Farklı öğretim yöntemlerinin öğrenci başarısı üzerindeki etkisini analiz etmek.

- Kalite Kontrol: Üretim hatalarının farklı nedenlerle ilişkisini incelemek.
- Genetik: Gen dağılımları ve özellikler arasındaki ilişkileri analiz etmek.

Ki-Kare Testinin Dikkat Edilmesi Gereken Noktaları ve Sınırlamaları:

- Kategorik Veri Gerekliliği: Ki-Kare testleri yalnızca kategorik (nominal veya ordinal) veriler için uygundur. Sürekli değişkenler için kullanılamaz.
- Beklenen Frekanslar: Beklenen frekansların çoğu yeterince büyük olmalıdır (genellikle her bir hücre için en az 5 beklenir). Düşük beklenen frekanslar, testin güvenirliğini azaltabilir. Bu durumda Fisher'ın kesin testi gibi alternatifler düşünülebilir.
- Bağımsız Gözlemler: Gözlemlerin birbirinden bağımsız olması gereklidir. Aynı bireyden birden fazla ölçüm alınması gibi durumlar testin varsayımlarını ihlal edebilir.
- İlişkinin Yönünü veya Gücünü Göstermez: Bağımsızlık testi, yalnızca değişkenler arasında bir ilişki olup olmadığını gösterir, bu ilişkinin yönü veya gücü hakkında bilgi vermez. İlişkinin gücünü ölçmek için Cramer's V veya Phi katsayısı gibi ek ölçüler kullanılabilir.
- Büyük Örneklem Gerekliliği: Ki-Kare testleri genellikle büyük örneklem boyutlarında daha güvenilirdir. Küçük örneklemlerde yanıltıcı sonuçlar verebilir.

3.2. Denetimsiz Öğrenme

Veri setinde sadece input bilgileri mevcut ise output bilgileri bulunmuyorsa yani etiket bilgisi yoksa bu öğrenme şekline denetimsiz öğrenme denir. Input bilgilerinin arasındaki benzerliklere göre kümeleme yapılır. Kümeleme problem tiplerinde denetimsiz öğrenme kullanılır. Örnek Algoritmalar:

- K-Ortalamalar Kümeleme (K-Means)**:** Verileri, önceden belirlenmiş sayıda kümeye ayırarak her küme içinde benzer veri noktalarını bir araya getirir.
- Bağlantısal Kümeleme (Hierarchical Clustering)**:** Verileri hiyerarşik bir yapıya göre kümeler.
- Temel Bileşenler Analizi (PCA)**:** Çok boyutlu veriyi daha az boyutlu hale getirerek veri boyutunu azaltmak için kullanılır.

3.2.1. Temel Bileşenler Analizi (PCA)

Temel Bileşenler Analizi (PCA), yüksek boyutlu veri setlerindeki karmaşıklığı azaltmak, verideki en önemli varyansı (bilgiyi) yakalamak ve bu sayede veriyi daha düşük boyutlu bir uzayda temsil etmek için kullanılan güçlü bir **boyut indirgeme** tekniğidir. PCA'nın temel amacı, orijinal değişkenler arasındaki korelasyonları dikkate alarak, birbirleriyle yüksek oranda ilişkili olan değişkenleri birleştirip, veri setinin ana yönlerini temsil eden yeni, birbiriyle ilişkisiz (ortogonal) değişkenler (temel bileşenler) oluşturmaktır.

PCA, denetimsiz bir öğrenme yöntemidir çünkü herhangi bir önceden tanımlanmış hedef değişken veya sınıf bilgisine ihtiyaç duymaz. Yalnızca veri setindeki özelliklerin kendisinden yola çıkarak yapıyı ve örüntüleri anlamaya çalışır.

PCA'nın Temel Mantığı: PCA'nın arkasındaki temel fikir, veri setindeki varyansı en iyi şekilde açıklayan doğrusal kombinasyonları (temel bileşenleri) bulmaktır. Varyans, verinin ne kadar yayıldığını veya dağıldığını gösterir. Yüksek varyans, veride daha fazla bilgi ve çeşitlilik olduğu anlamına gelir. PCA, en yüksek varyansı taşıyan yönleri (temel bileşenleri) ilk sıraya yerleştirir ve sonraki bileşenler, önceki bileşenlerle ilişkisiz olacak şekilde, kalan varyansı en iyi şekilde açıklamaya çalışır. PCA genellikle aşağıdaki adımları içerir.

Veri Standardizasyonu (Ölçeklendirme): Orijinal veri setindeki değişkenlerin farklı ölçeklerde olması, PCA sonuçlarını olumsuz etkileyebilir. Bu nedenle, genellikle her bir değişkenin ortalaması sıfıra ve standart sapması bire eşit olacak şekilde standartlaştırılır. Bu işlem, tüm değişkenlerin analize eşit katkıda bulunmasını sağlar. Matematiksel olarak, bir xi değeri için standartlaştırılmış değer zi=σxi−μ şeklinde hesaplanır, burada μ değişkenin ortalaması ve σ standart sapmasıdır.

Kovaryans Matrisinin Hesaplanması: Standardize edilmiş veri setindeki tüm değişken çiftleri arasındaki kovaryansı gösteren bir kovaryans matrisi hesaplanır. Kovaryans, iki değişkenin birlikte ne kadar değiştiğinin bir ölçüsüdür. Pozitif kovaryans, değişkenlerin aynı yönde hareket ettiğini, negatif kovaryans ise ters yönde hareket ettiğini gösterir. Kovaryans matrisinin boyutu, değişken sayısı x değişken sayısıdır.

Özvektörlerin ve Özdeğerlerin Hesaplanması: Kovaryans matrisinin özvektörleri (eigenvectors) ve özdeğerleri (eigenvalues) hesaplanır.

- Özvektörler: Verideki temel yönleri veya eksenleri temsil eden vektörlerdir. Her bir özvektör, orijinal değişkenlerin bir doğrusal kombinasyonunu ifade eder.
- Özdeğerler: Her bir özvektörle ilişkilendirilen ve o özvektörün temsil ettiği varyansın büyüklüğünü gösteren skaler değerlerdir. Büyük bir özdeğer, ilgili özvektörün veri setindeki varyansın önemli bir bölümünü açıkladığı anlamına gelir.

Özvektörlerin Özdeğerlere Göre Sıralanması: Özvektörler, karşılık gelen özdeğerlerin büyüklüğüne göre azalan sırada sıralanır. En yüksek özdeğere sahip olan özvektör (birinci temel bileşen), veri setindeki en büyük varyansı temsil eder. İkinci en yüksek özdeğere sahip olan özvektör (ikinci temel bileşen), birinci bileşenle ilişkisiz olacak şekilde kalan varyansı en iyi şekilde açıklar ve bu şekilde devam eder.

Temel Bileşenlerin Seçilmesi (Boyutun Belirlenmesi): Amaçlanan boyut indirgeme seviyesine bağlı olarak, en yüksek özdeğerlere sahip olan ilk k sayıda özvektör (temel bileşen) seçilir. k, orijinal değişken sayısından daha küçüktür. k değerinin seçimi genellikle açıklanan varyans oranı, dirsek metodu (scree plot) veya uygulama gereksinimleri gibi çeşitli kriterlere göre yapılır.

Açıklanan varyans oranı, seçilen temel bileşenlerin toplam varyansın ne kadarını açıkladığını gösterir. Genellikle, toplam varyansın belirli bir yüzdesini (örneğin %90 veya %95) açıklayan sayıda bileşen seçilir.

Verinin Yeni Temel Bileşenlere Projeksiyonu: Orijinal veri seti, seçilen k adet temel bileşen (özvektör) ile çarpılarak daha düşük boyutlu bir uzaya yansıtılır. Bu işlem, her bir veri noktasının yeni temel bileşenler üzerindeki skorlarını (değerlerini) elde etmemizi sağlar. Elde edilen yeni veri seti, orijinal verinin daha düşük boyutlu bir temsilidir ve ana varyansı korur.

PCA'nın Faydaları ve Kullanım Alanları:

- Boyut İndirgeme: Yüksek boyutlu veriyi daha az sayıda değişkene indirgeyerek veri analizini, görselleştirmesini ve modellemesini kolaylaştırır.

- Veri Görselleştirme: Veriyi 2 veya 3 boyutta temsil ederek karmaşık ilişkilerin ve örüntülerin anlaşılmasını sağlar.
- Özellik Mühendisliği: Makine öğrenmesi modelleri için daha az sayıda, ancak daha anlamlı ve birbiriyle ilişkisiz özellikler elde edilmesine yardımcı olur. Bu, modelin performansını artırabilir ve aşırı uyumu azaltabilir.
- Gürültü Azaltma: Daha az varyans taşıyan temel bileşenlerin atılmasıyla verideki gürültü ve gereksiz bilginin filtrelenmesine yardımcı olabilir.
- Veri Sıkıştırma: Veriyi daha az yer kaplayacak şekilde temsil etmeyi mümkün kılar.
- Uygulama Alanları: Görüntü işleme (yüz tanıma), biyoinformatik (genomik veri analizi), finans (risk yönetimi), doğal dil işleme (metin analizi) ve daha birçok alanda yaygın olarak kullanılır.

PCA'nın Dikkat Edilmesi Gereken Noktaları ve Sınırlamaları:

- Doğrusal İlişkiler: PCA, değişkenler arasındaki doğrusal ilişkileri yakalamada etkilidir. Doğrusal olmayan ilişkiler söz konusu olduğunda performansı düşebilir. Bu tür durumlarda, çekirdek PCA (Kernel PCA) gibi doğrusal olmayan boyut indirgeme teknikleri düşünülebilir.
- Ölçeklendirme Hassasiyeti: PCA, değişkenlerin ölçeklerinden önemli ölçüde etkilenir. Bu nedenle, verinin doğru bir şekilde ölçeklendirilmesi kritik öneme sahiptir.
- Yorumlanabilirlik Kaybı: Orijinal değişkenlerin doğrusal kombinasyonları olan temel bileşenlerin yorumlanması bazen zor olabilir. Her bir temel bileşenin hangi orijinal değişkenlerden ne kadar etkilendiğini anlamak için "yükler" (loadings) incelenebilir.
- Varyans Her Zaman Önemli Bilgi Demek Değildir: PCA, en yüksek varyansı korumaya odaklanır. Ancak, en yüksek varyansı taşıyan bileşenler her zaman problem için en anlamlı veya en prediktif olanlar olmayabilir.

3.2.2. Kmeans

K-means, veri biliminde kullanılan popüler bir kümeleme algoritmasıdır. Temel amacı, bir veri kümesini önceden belirlenmiş sayıda kümeye (K) ayırmaktır. Bu algoritma, verileri benzer özelliklere sahip gruplar halinde düzenleyerek, veriler arasındaki gizli yapıları ortaya çıkarmaya yardımcı olur. İşte K-means algoritmasının temel prensipleri ve nasıl çalıştığına dair bazı bilgiler:

Temel Kavramlar:

- Kümeleme: Veri noktalarını benzer özelliklere göre gruplara ayırma işlemidir.
- K: Oluşturulacak küme sayısıdır ve algoritmanın başlangıcında belirlenir.
- Merkez Noktaları (Centroids): Her bir kümeyi temsil eden ve küme içindeki veri noktalarının ortalamasını ifade eden noktalardır.
- Uzaklık Ölçüsü: Veri noktaları arasındaki benzerliği veya farklılığı ölçmek için kullanılan bir metriktir (genellikle Öklid mesafesi kullanılır).

K-means Algoritmasının Adımları:

Başlangıç Merkez Noktalarının Belirlenmesi:

- Rastgele olarak K adet merkez noktası seçilir.

Veri Noktalarının Kümelere Atanması:

- Her bir veri noktası, en yakın merkez noktasına atanır.

Merkez Noktalarının Güncellenmesi:

- Her bir küme için, küme içindeki veri noktalarının ortalaması alınarak yeni merkez noktaları hesaplanır.

Tekrarlama:

- 2 ve 3. adımlar, merkez noktalarında veya küme atamalarında herhangi bir değişiklik olmayana kadar tekrarlanır.

K-means'in Kullanım Alanları:

- Müşteri Segmentasyonu: Müşterileri satın alma alışkanlıklarına veya demografik özelliklerine göre gruplandırma.
- Görüntü İşleme: Görüntüleri renk veya doku benzerliklerine göre segmentlere ayırma.

- Belge Kümeleme: Belgeleri konu veya temalarına göre gruplandırma.
- Anomali Tespiti: Aykırı veya sıra dışı veri noktalarını belirleme.

K-means'in Avantajları:

- Uygulaması kolay ve hızlıdır.
- Büyük veri kümeleriyle etkili bir şekilde çalışabilir.

K-means'in Dezavantajları:

- K küme sayısının önceden belirlenmesi gerekir.
- Başlangıç merkez noktalarına duyarlıdır.
- Gürültülü veya aykırı verilere karşı hassastır.
- Küresel olmayan küme şekillerinde iyi sonuç vermeyebilir.

K-means, veri analizinde güçlü bir araçtır ve birçok farklı uygulama alanında kullanılmaktadır. Ancak, algoritmanın sınırlamalarını ve veri kümesinin özelliklerini dikkate almak önemlidir.

3.3. Yarı Denetimli Öğrenme

Bazı veri setlerinde veriler gürültülü, eksik, boş olabilir. Bazı input'ların output bilgileri mevcutken bazılarının mevcut olmayabilir. Dolayısıyla ilk önce verilere kümeleme uygulanarak verilere etiket bilgisi kazandırılır. Daha sonra da sınıflandırma yapılarak mevcutta output'u olmayan input'lara karşılık gelen output'ları ortaya konur.

3.4. Takviyeli Öğrenme

Denetimli öğrenmede olduğu gibi input ve bu input'lara karşılık gelen output bilgileri veri setinde mevcuttur. Fakat takviyeli öğrenme sadece input değerleriyle ilgilenir. Input'lara göre bir output üretir. Olması gereken output ile üretilen output karşılaştırılır. Eğer sonuçlar birbirine yakınsa ödüllendirme yapılır. Ödüllendirme ise şudur: bağlantıların matematiksel değerleri daha büyük derecelendirilir.

4. DERİN ÖĞRENME

Derin öğrenme (Deep Learning), yapay zekanın (YZ) bir alt kümesi olan makine öğrenmesinin (Machine Learning) bir dalıdır. Temelinde, insan beyninin yapısından esinlenerek tasarlanmış çok katmanlı yapay sinir ağları (Artificial Neural Networks - ANN) bulunur. Bu ağlar, karmaşık veri örüntülerini ve soyutlamalarını otomatik olarak öğrenme yeteneğine sahiptir. "Derin" terimi, geleneksel sinir ağlarına kıyasla çok daha fazla sayıda gizli katmana sahip olmasından gelir.

Derin Öğrenmenin Temel Farkları ve Avantajları:

- Otomatik Özellik Çıkarımı (Automatic Feature Extraction): Geleneksel makine öğrenmesi algoritmalarında, veriden anlamlı özellikleri (örneğin, bir görüntüdeki köşe noktaları, bir metindeki kelime sıklığı) elle belirlemek ve çıkarmak gerekebilir. Derin öğrenme modelleri ise, çok sayıda katman sayesinde ham veriden (örneğin, piksel değerleri, metin karakterleri) doğrudan hiyerarşik bir şekilde karmaşık özellikleri otomatik olarak öğrenirler. Bu, özellikle yapılandırılmamış veriler (görüntüler, ses, metin) için büyük bir avantaj sağlar.
- Karmaşık İlişkileri Modelleme Yeteneği: Derin sinir ağlarının çok katmanlı yapısı, veriler arasındaki doğrusal olmayan ve karmaşık ilişkileri etkili bir şekilde modelleyebilir. Bu sayede, geleneksel algoritmaların zorlandığı karmaşık problemleri çözme potansiyeline sahiptir.
- Büyük Veriyle Daha İyi Performans: Derin öğrenme modelleri, genellikle büyük miktarda veriyle eğitildikçe performanslarını önemli ölçüde artırırlar. Geleneksel algoritmalar ise belirli bir veri büyüklüğünden sonra performanslarında doygunluğa ulaşabilirler.
- Uçtan Uca Öğrenme (End-to-End Learning): Bazı durumlarda, derin öğrenme modelleri ham girdiden doğrudan nihai çıktıya kadar tüm süreci tek bir model içinde öğrenebilirler. Bu, ara adımlardaki manuel özellik mühendisliği ihtiyacını ortadan kaldırır.

Derin Öğrenmenin Temel Bileşenleri:

- Yapay Sinir Ağları (Artificial Neural Networks - ANN): Derin öğrenmenin temel yapı taşıdır. Biyolojik sinir hücrelerinden esinlenerek tasarlanmış birbirine bağlı düğümlerden (nöronlar) oluşur. Her bağlantı, bir ağırlığa sahiptir ve bu ağırlıklar eğitim sürecinde ayarlanarak ağın öğrenmesi sağlanır.
- Katmanlar (Layers): Sinir ağları, genellikle giriş katmanı (input layer), bir veya daha fazla sayıda gizli katman (hidden layers) ve bir çıkış katmanı (output layer) içerir. Derin öğrenme ağlarında çok sayıda gizli katman bulunur. Her katman, bir önceki katmanın çıktısını girdi olarak alır ve belirli bir dönüşüm uygular.
- Nöronlar (Neurons): Her katmandaki temel işlem birimidir. Girdileri alır, ağırlıklarla çarpar, bir toplama işlemi yapar ve ardından bir aktivasyon fonksiyonu uygular. Aktivasyon fonksiyonu, nöronun çıktısını belirler ve ağa doğrusal olmayanlık katar.
- Ağırlıklar (Weights) ve Biaslar (Biases): Nöronlar arasındaki bağlantıların gücünü temsil eden parametrelerdir. Eğitim sürecinde, modelin tahminleri ile gerçek değerler arasındaki hatayı en aza indirecek şekilde ayarlanırlar. Biaslar ise nöronların aktivasyon eşiğini kontrol eder.
- Aktivasyon Fonksiyonları (Activation Functions): Nöronların çıktısını belirleyen ve ağa doğrusal olmayanlık katan matematiksel fonksiyonlardır. Yaygın kullanılan aktivasyon fonksiyonları arasında sigmoid, ReLU (Rectified Linear Unit), tanh ve softmax bulunur.
- Kayip Fonksiyonu (Loss Function) / Maliyet Fonksiyonu (Cost Function): Modelin tahminleri ile gerçek değerler arasındaki farkı ölçen bir fonksiyondur. Eğitim sürecinin amacı, bu kaybı en aza indirmektir.
- Optimizasyon Algoritmaları (Optimization Algorithms): Kayıp fonksiyonunu en aza indirmek için modelin ağırlıklarını ve biaslarını nasıl güncelleyeceğini belirleyen algoritmalardır. Gradyan inişi (Gradient Descent) ve onun türevleri (örneğin, Adam, RMSprop) en sık kullanılan optimizasyon algoritmalarındandır.

Derin Öğrenme Modellerinin Türleri:

Derin öğrenme, farklı veri türleri ve problem türleri için çeşitli mimarilere sahip sinir ağları kullanır:

- Evrişimsel Sinir Ağları (Convolutional Neural Networks - CNNs): Özellikle görüntü ve video işleme görevlerinde başarılıdır. Evrişim (convolution) katmanları, pooling katmanları ve tam bağlantılı (fully connected) katmanlardan oluşurlar. Görüntülerdeki yerel örüntüleri (kenarlar, köşeler, nesne parçaları) otomatik olarak öğrenirler.
- Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNNs): Ardışık (sequential) verileri (örneğin, metin, zaman serileri, ses) işlemek için tasarlanmıştır. Geriye dönük bağlantıları sayesinde önceki girdiler hakkında bilgi saklayabilirler. Ancak uzun süreli bağımlılıkları öğrenmede zorluk çekebilirler.
- Uzun Kısa Süreli Bellek Ağları (Long Short-Term Memory Networks - LSTMs) ve Kapılı Tekrarlayan Birimler (Gated Recurrent Units - GRUs): RNN'lerin uzun süreli bağımlılıkları öğrenme sorununu çözmek için geliştirilmiş daha karmaşık tekrarlayan ağ mimarileridir. Bellek hücreleri ve kapılar (gates) aracılığıyla bilgiyi daha etkili bir şekilde saklayabilir ve yönetebilirler.
- Transformatör Ağları (Transformer Networks): Özellikle doğal dil işleme (NLP) alanında devrim yaratmıştır. Dikkat mekanizmalarını (attention mechanisms) kullanarak girdideki farklı pozisyonlar arasındaki ilişkileri modelleyebilirler. Paralel işleme yetenekleri sayesinde uzun dizileri etkili bir şekilde işleyebilirler. BERT, GPT gibi popüler dil modelleri transformatör mimarisini temel alır.
- Üretken Çekişmeli Ağlar (Generative Adversarial Networks - GANs): Üretken modeller (generator) ve ayrımcı modellerden (discriminator) oluşan iki ağın birbirleriyle rekabet ederek veri dağılımını öğrenmesini sağlayan bir yaklaşımdır. Yeni ve gerçekçi veri örnekleri (görüntüler, metinler vb.) üretmek için kullanılırlar.

- Otomatik Kodlayıcılar (Autoencoders): Girdiyi daha düşük boyutlu bir temsile (kod) sıkıştırmayı ve ardından bu kottan orijinal girdiyi yeniden oluşturmayı öğrenen ağlardır. Özellik öğrenimi, boyut indirgeme ve anomali tespiti gibi amaçlarla kullanılabilirler.

Derin Öğrenmenin Uygulama Alanları:

Derin öğrenme, günümüzde birçok farklı alanda çığır açan uygulamalara olanak sağlamıştır:

- Bilgisayarla Görü (Computer Vision): Görüntü tanıma, nesne tespiti, görüntü segmentasyonu, yüz tanıma, görüntü oluşturma, video analizi.
- Doğal Dil İşleme (Natural Language Processing - NLP): Metin sınıflandırma, duygu analizi, makine çevirisi, soru cevaplama, metin özetleme, dil üretimi, sohbet botları.
- Ses Tanıma (Speech Recognition): Konuşmayı metne dönüştürme.
- Öneri Sistemleri (Recommender Systems): Kullanıcılara ilgi alanlarına uygun ürün veya içerik önerme.
- Sağlık Hizmetleri (Healthcare): Tıbbi görüntü analizi, hastalık teşhisi, ilaç keşfi.
- Finans (Finance): Dolandırıcılık tespiti, risk yönetimi, algoritmik ticaret.
- Otonom Araçlar (Autonomous Vehicles): Çevre algılama, yol takibi, trafik işareti tanıma.
- Robotik (Robotics): Robotların çevreleriyle etkileşim kurması ve karmaşık görevleri yerine getirmesi.

Derin Öğrenmenin Zorlukları ve Dikkat Edilmesi Gerekenler:

- Büyük Veri İhtiyacı: Derin öğrenme modelleri genellikle çok büyük miktarda etiketlenmiş veriyle eğitilmeyi gerektirirler.
- Yüksek Hesaplama Kaynakları: Derin modellerin eğitimi ve çalıştırılması önemli miktarda işlem gücü (GPU'lar) ve bellek gerektirebilir.
- Yorumlanabilirlik Sorunu (Black Box): Derin öğrenme modellerinin nasıl karar verdiğini anlamak genellikle zordur. Bu durum, özellikle kritik uygulamalarda (örneğin, sağlık, hukuk) bir dezavantaj olabilir.

- Hiperparametre Ayarı: Derin öğrenme modellerinin performansını etkileyen birçok hiperparametre (örneğin, katman sayısı, nöron sayısı, öğrenme oranı) vardır ve bunların doğru şekilde ayarlanması zaman alıcı ve deneyim gerektiren bir süreç olabilir.
- Aşırı Uyum (Overfitting): Modelin eğitim verisine çok fazla uyum sağlayıp, yeni ve görülmemiş verilerde kötü performans gösterme riski vardır. Düzenlileştirme teknikleri (örneğin, dropout, weight decay) bu sorunu azaltmaya yardımcı olabilir.

Sonuç olarak, derin öğrenme, yapay zeka alanında önemli bir atılımdır ve karmaşık veri örüntülerini otomatik olarak öğrenme yeteneği sayesinde birçok alanda çığır açan uygulamalara olanak sağlamaktadır. Sürekli gelişen algoritmaları, artan hesaplama gücü ve büyük veri kaynaklarının yaygınlaşmasıyla birlikte, derin öğrenmenin gelecekte de hayatımızın birçok yönünü derinden etkilemesi beklenmektedir.

5. DUYGU ANALİZİ

Duygu analizi (Sentiment Analysis), metinlerde ifade edilen duygu, görüş, düşünce ya da tutumu otomatik olarak tanımlayan bir Doğal Dil İşleme (Natural Language Processing - NLP) tekniğidir. Bu yöntem, metinlerdeki pozitif, negatif veya nötr duyguları anlamayı ve kategorize etmeyi hedefler. Genellikle müşteri geri bildirimleri, sosyal medya içerikleri, incelemeler veya diğer metinsel veri türleri üzerinde kullanılır.

Duygu Analizi Süreçleri

Veri Toplama: Metin madenciliğinin temel verisini metinler oluşturmaktadır. Veri setini oluşturmak için ilgilenilen konuya dair metinlere ulaşmak ilk aşamadır. Günümüzde en önemli metin kaynağı olarak internet karşımıza çıkmaktadır. Google, Yandex gibi arama motorları ile Twitter, Facebook gibi sosyal ağlar en temel kaynaklar olarak görülebilir. Aynı şekilde kurumların oluşturmuş oldukları kendi veri ambarları, yazılmış raporlar çevirim içi araçlar da veri seti oluşturabilirler. Teknolojinin bize sağlamış olduğu büyük nimetlerden biri de veri setlerine rahatlıkla ulaşabilmektir. Bu konuda hali hazırda var olan veri setleri kullanılabileceği gibi internet ortamından anlık veri çekebilmek amacıyla geliştirilmiş olan paket programlar da kullanılabilir. Bazı firmalar ise kendi derledikleri verileri açık kullanıma sunmakta, bireyler veri setine erişerek bu verileri alıp kullanma şansına sahip olmaktadırlar.

Metin Ön İşleme: Veri setinde gerçekleştirilecek ilk aşama, veri ön işleme sürecidir. Genellikle toplanan metin verileri, doğrudan analiz için uygun bir yapıda olmaz. Bu nedenle metinlerin, kullanılabilir hale getirilmesi amacıyla çeşitli işlemlerden geçirilmesi gerekir. Kelimelerin ayrıştırılması, anlamsal içeriklerinin belirlenmesi, köklerine indirgenmesi, gereksiz kelimelerin ve anlam taşımayan ifadelerin temizlenmesi, yazım ve imla hatalarının düzeltilmesi gibi işlemler, ön işleme sürecinin temel adımları arasında yer alır. Metin madenciliğinde kullanılan dokümanlar çok farklı biçimlerde ve düzensiz içeriklerde olabilir. Bu durum, metinlerin analiz için uygun hale getirilmesi adına gelişmiş yöntemler geliştirilmesini gerekli kılar. Ancak bu işlemlerin detaylı olması, genellikle zaman alıcı bir süreç ortaya çıkarır.

Bu nedenle, analizden güvenilir ve doğru sonuçlar elde edebilmek için ön işleme adımlarının dikkatle yürütülmesi büyük önem taşır. Bu süreç, aşağıda özetlenen temel işlemleri kapsamaktadır (Kızılkaya, 2018, s. 8).

Tokenization: Metin kelimelere veya ifadelere bölünür. Metin işleme sürecinde atılması gereken ilk adım, etiket ve özel karakter temizleme işlemidir. Bu adım, karakter dizilerinden oluşan ham metnin, makine öğrenmesi algoritmalarının işleyebileceği uygun bir yapıya dönüştürülmesini amaçlar. Özellikle web kaynaklı verilerle çalışıldığında, metin içinde sıklıkla karşılaşılan XML (Extensible Markup Language) ve HTML (Hyper Text Markup Language) gibi işaretleme dilleriyle eklenmiş etiketlerin temizlenmesi gerekir. Bu etiketler, web sayfalarında yapısal anlam taşısa da, metin analizi açısından anlamsızdır ve veri bütünlüğüne katkı sağlamaz. Aynı şekilde, noktalama işaretleri, satır sonları ve diğer okunamayan ya da analiz açısından gereksiz olan karakterler, genellikle boşluk karakteriyle değiştirilerek temizlenir. Bu işlemler sonucunda, analiz sürecine katkısı bulunmayan kısaltmalar, semboller ve işaretlemeler metinden ayıklanmış olur (Kızılkaya, 2018, s. 9).

Lemmatization/Stemming: Metin işleme sürecinin önemli aşamalarından biri de kelimelerin köklerine indirgenmesidir. Kök, ek almadan da anlam ifade edebilen en temel kelime formudur. Örneğin, "gel" veya "git" gibi sözcükler, yalın halleriyle anlamlıdır. Bu işlemin amacı, kelimelerin kök formlarını tespit ederek metindeki çeşitliliği azaltmak ve analizi daha tutarlı hale getirmektir. Türkçe dilinin sondan eklemeli yapısı nedeniyle, aynı kökten türeyen birçok farklı kelime formu ortaya çıkabilir. Örneğin, "evler" kelimesi köküne indirgendikten sonra "ev" olur ya da "okuyor" kelimesi "oku" köküne indirgenir. Böylece çoğul ekleri ve çekim ekleri gibi yapılar temizlenmiş olur. Ancak kök bulma işlemi bazı zorlukları da beraberinde getirebilir. Yanlış bir uygulama sonucunda, kelimenin kökü olması gereken yerden fazla ayrıştırılarak anlam kaybı yaşanabilir ya da köke ulaşmak için yeterince işlem yapılmadığında istenilen sonuç elde edilemeyebilir. Bu nedenle kök bulma işleminin dil bilgisi kurallarına uygun bir şekilde yapılması büyük önem taşır. Türkçede kök bulma sırasında harf düşmesi, ünlü uyumu, kaynaştırma harfleri gibi dilin yapısal özellikleriyle de karşılaşılabilir.

Yanlış yapılan kök bulma işlemleri, metnin anlam bütünlüğünü bozarak analiz kalitesini olumsuz etkileyebilir. Türkçe metinler için bu işlemi destekleyen açık kaynaklı araçlardan biri Zemberek dil işleme kütüphanesidir. Zemberek, kelimeleri köklerine ayırma işlemini dil bilgisi kurallarına uygun olarak gerçekleştirmeye olanak sağlar ve bu sayede analiz sürecinde önemli bir yardımcı araç olarak kullanılabilir (Kızılkaya, 2018, s. 9).

Part-of-Speech (POS) Tagging: Metin işleme sürecinde bir diğer önemli adım, kelimelerin dil bilgisel türlerinin belirlenmesi yani hangi kelimenin isim, fiil, sıfat gibi bir kategoriye ait olduğunun tespit edilmesidir. Bu işlem tamamlandıktan sonra, cümlenin genel anlamını etkilemeyen ve genellikle bilgi değeri taşımayan "durak kelimeler" (stop words) metinden çıkarılır. Durak kelimeler; edatlar, bağlaçlar ve zamirler gibi, metnin anlam bütünlüğüne doğrudan katkıda bulunmayan ancak dilde sıkça kullanılan ifadelerdir. Bu tür kelimelerin metinden temizlenmesi, analiz sürecinde hem gereksiz işlem yükünü azaltır hem de modelin daha anlamlı verilere odaklanmasını sağlar. Örneğin, Türkçede çok sık geçen "ile", "çünkü", "gibi", "sadece" gibi sözcüklerin metinden çıkarılması, işlem görecek kelime sayısını önemli ölçüde azaltarak hem işlem süresini kısaltır hem de model performansını artırabilir. Bu nedenle, analiz öncesinde durak kelimelerin dikkatlice temizlenmesi, daha doğru ve verimli sonuçlar elde etmek açısından büyük önem taşımaktadır.

Duygu Kategorisi Belirleme:

- Belirli duygu kategorileri (pozitif, negatif, nötr vb.) tanımlanır.
- Metindeki duygu eğilimleri, kullanılan yöntemlere göre sınıflandırılır.

Sonuçların Analizi:

- Belirlenen duygular, genel eğilimleri anlamak için değerlendirilir ve görselleştirilir.

Duygu Analizi Seviyeleri

Metin Seviyesi (Document-Level Analysis):

- Bir metnin tamamının genel duygu yönelimini belirler.
- Örneğin: "Bu restoran harika bir yemek deneyimi sunuyor." → Pozitif

Cümle Seviyesi (Sentence-Level Analysis):

- Her cümlenin duygu polaritesini belirler.
- Örneğin: "Servis çok hızlıydı, ama yemekler soğuktu."
  - "Servis çok hızlıydı." → Pozitif
  - "Ama yemekler soğuktu." → Negatif

Kelime/Özellik Seviyesi (Aspect-Based Sentiment Analysis):

- Metindeki belirli konular veya özelliklere yönelik duyguları analiz eder.
- Örneğin: "Telefonun kamerası çok iyi, ancak pil ömrü kötü."
  - Kamera → Pozitif
  - Pil ömrü → Negatif

Duygu Analizi Uygulamaları

Müşteri Geri Bildirimi Analizi: Şirketler, müşteri memnuniyetini ölçmek için ürün veya hizmet incelemelerini analiz eder.

Sosyal Medya İzleme**:** Markalar, Twitter, Instagram gibi platformlarda kullanıcıların duygusal eğilimlerini analiz eder.

Film ve Ürün İnceleme Analizi: Bir film, kitap ya da ürün hakkındaki genel kullanıcı görüşlerini anlamak için kullanılır.

Siyasi Analiz: Kamuoyunun belirli politik konulara veya liderlere karşı tutumunu analiz etmek için kullanılır.

Haber Analizi: Haber içeriklerinin pozitif ya da negatif duygu eğilimlerini analiz etmek için kullanılır.

Kullanılan Teknikler

Kural Tabanlı Yöntemler

- Dilbilgisi kuralları ve kelime listeleri kullanılır.
- Kelimelere pozitif, negatif veya nötr skorlar atanır.
- Basit ve hızlıdır, ancak bağlamı anlamada yetersizdir.

Makine Öğrenmesi Tabanlı Yöntemler

- Veri etiketlenir ve bir model eğitilir.
- Algoritmalar: Destek Vektör Makineleri (SVM), Naive Bayes, Random Forest (RF), Lojistik Regresyon.
- Daha yüksek doğruluk sağlar, ancak büyük miktarda etiketli veriye ihtiyaç duyar.

Derin Öğrenme Tabanlı Yöntemler

- Sinir ağları (RNN, LSTM, Transformer) kullanılarak bağlamı daha iyi anlayan modeller geliştirilir.
- Örneğin: BERT, GPT gibi modeller.
- Daha karmaşıktır ve güçlü donanım gerektirir.

6. PYTHON

Python, Guido van Rossum tarafından geliştirilen ve 1991'de kullanıma sunulan, yüksek seviyeli, yorumlanabilir, dinamik tipli ve genel amaçlı bir programlama dilidir. Temel felsefesi, kodun okunabilirliğini ön planda tutmak ve geliştiricilere az kodla güçlü çözümler üretme imkanı sunmaktır. İngilizceye yakın söz dizimi ve zengin standart kütüphanesi sayesinde hem öğrenmesi kolay hem de geniş bir uygulama yelpazesine hitap eden bir araçtır.

Python'ın Geniş Kullanım Alanları: Python'ın çok yönlülüğü, onu sayısız alanda tercih edilen bir dil haline getirmiştir:

- Web Geliştirme: Django ve Flask gibi güçlü ve popüler web framework'leri sayesinde karmaşık ve ölçeklenebilir web uygulamaları, API'ler ve web servisleri geliştirmede yaygın olarak kullanılır. Büyük platformlardan küçük projelere kadar geniş bir yelpazede uygulama geliştirme imkanı sunar.
- Veri Bilimi ve Makine Öğrenmesi: NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch, Keras gibi güçlü kütüphaneleri sayesinde veri analizi, veri görselleştirme, istatistiksel modelleme ve makine öğrenmesi algoritmalarının geliştirilmesi ve uygulanmasında lider konumdadır. Büyük veri setleriyle çalışma, karmaşık analizler yapma ve yapay zeka uygulamaları geliştirme imkanı sunar.
- Bilimsel Hesaplama ve Mühendislik: SciPy, Matplotlib gibi kütüphanelerle bilimsel araştırmalar, mühendislik simülasyonları, matematiksel modelleme ve veri görselleştirme gibi alanlarda kullanılır.
  Akademik araştırmalardan endüstriyel uygulamalara kadar geniş bir yelpazede bilimsel problemlerin çözümüne katkıda bulunur.
- Otomasyon ve Scripting: Sistem yönetimi görevlerini otomatikleştirmek, dosya işlemleri yapmak, tekrarlayan görevleri otomatikleştiren scriptler yazmak için idealdir. Küçük yardımcı programlardan karmaşık otomasyon çözümlerine kadar çeşitli amaçlarla kullanılabilir.

- Oyun Geliştirme: Pygame gibi kütüphanelerle basit 2D oyunlar ve prototipler geliştirmek mümkündür. Büyük oyun motorları olmasa da, oyun geliştirme prensiplerini öğrenmek ve basit oyun projeleri oluşturmak için uygun bir platform sunar.
- Masaüstü Uygulamaları: Tkinter, PyQt gibi kütüphaneler aracılığıyla platform bağımsız grafiksel kullanıcı arayüzüne (GUI) sahip masaüstü uygulamaları geliştirmek mümkündür. İş araçlarından özel amaçlı uygulamalara kadar çeşitli masaüstü yazılımları oluşturulabilir.
- Ağ Programlama: Socket programlama ve çeşitli ağ kütüphaneleri sayesinde ağ uygulamaları, istemci-sunucu sistemleri ve ağ protokolleri ile etkileşimli uygulamalar geliştirmek mümkündür.
- Eğitim: Öğrenmesi kolay ve anlaşılır söz dizimi sayesinde programlama eğitiminde başlangıç dili olarak sıklıkla tercih edilir. Temel programlama kavramlarını öğretmek ve öğrencilerin programlamaya kolayca adapte olmasını sağlamak için ideal bir dildir.
- Diğer Alanlar: Yapay zeka, doğal dil işleme (NLP), görüntü işleme, siber güvenlik, biyoinformatik gibi birçok farklı alanda da Python aktif olarak kullanılmaktadır.

Python'ın Avantajları:

- Öğrenme Kolaylığı ve Okunabilirlik: Python'ın söz dizimi basit, düzenli ve İngilizceye yakındır. Bu, yeni başlayanların dili hızlı bir şekilde öğrenmesini ve mevcut kodu kolayca anlamasını sağlar. Okunabilirlik, ekip çalışmalarında ve kodun sürdürülebilirliğinde önemli bir avantajdır.
- Geniş ve Aktif Topluluk: Büyük ve aktif bir geliştirici topluluğu sayesinde bol miktarda kaynak (dokümantasyon, eğitimler, forumlar) bulunur. Karşılaşılan sorunlara çözüm bulmak ve yeni bilgiler edinmek kolaydır.
- Zengin Standart Kütüphane: Python, birçok farklı görevi gerçekleştirmek için hazır fonksiyonlar ve modüller içeren kapsamlı bir standart kütüphaneye sahiptir. Bu, temel işlemleri gerçekleştirmek için ek kütüphanelere olan ihtiyacı azaltır ve geliştirme sürecini hızlandırır.

- Geniş Üçüncü Taraf Kütüphane Ekosistemi: Veri bilimi, makine öğrenmesi, web geliştirme gibi özel alanlarda çok sayıda güçlü ve olgunlaşmış üçüncü taraf kütüphane bulunur. Bu kütüphaneler, karmaşık problemleri çözmek için hazır araçlar sunar ve geliştirme sürecini önemli ölçüde hızlandırır.
- Çoklu Platform Desteği: Python kodu, Windows, macOS, Linux gibi farklı işletim sistemlerinde değişiklik yapmadan çalışabilir. Bu, platform bağımsız uygulama geliştirme imkanı sunar.
- Dinamik Tiplendirme: Değişkenlerin tiplerinin çalışma anında belirlenmesi, hızlı prototipleme ve esnek kod yazma imkanı sunar.
- Nesne Yönelimli ve Fonksiyonel Programlama Desteği: Hem nesne yönelimli hem de fonksiyonel programlama paradigmalarını destekleyerek farklı programlama yaklaşımlarını kullanma esnekliği sunar.
- Yorumlanabilir Dil: Kodun derlenmeye ihtiyaç duymadan doğrudan çalıştırılması, geliştirme ve test süreçlerini hızlandırır.

Python'ın Dezavantajları:

- Hız: Yorumlanabilir bir dil olduğu için, derlenen dillere (örneğin C++, Java) kıyasla bazı durumlarda daha yavaş çalışabilir. Performans kritik uygulamalarda bu bir dezavantaj olabilir. Ancak, performans gerektiren kısımlar genellikle C/C++ gibi dillerde yazılmış kütüphanelerle desteklenir.
- Bellek Tüketimi: Dinamik tiplendirme ve otomatik bellek yönetimi nedeniyle bazı durumlarda derlenen dillere göre daha fazla bellek tüketebilir.
- Global Interpreter Lock (GIL): CPython'da (en yaygın Python yorumlayıcısı), aynı anda yalnızca bir thread'in Python bytecode'unu çalıştırmasına izin veren bir mekanizma olan GIL bulunur.

  Bu, çok çekirdekli işlemcilerden tam olarak yararlanmayı zorlaştırabilir ve bazı paralel işleme senaryolarında performans sınırlamalarına yol açabilir. Ancak, multiprocessing modülü gibi araçlarla bu sınırlamaların üstesinden gelinebilir.

- Mobil Geliştirme: Web geliştirme ve masaüstü uygulamaları için güçlü olsa da, saf Python ile yüksek performanslı mobil uygulamalar geliştirmek diğer platformlara (örneğin Android için Kotlin/Java, iOS için Swift/Objective-C) kıyasla daha sınırlıdır. Ancak, Kivy gibi framework'ler mobil uygulama geliştirmek için kullanılabilir.
- Çalışma Zamanı Hataları: Dinamik tiplendirme nedeniyle, tip hataları derleme zamanında değil, çalışma zamanında ortaya çıkabilir. Bu, daha dikkatli test yapmayı gerektirebilir.

6.1. Pandas

Pandas, Python programlama dili için geliştirilmiş, açık kaynaklı ve yüksek performanslı bir veri analizi ve manipülasyon kütüphanesidir. Wes McKinney tarafından 2008 yılında geliştirilmeye başlanmış olup, günümüzde veri bilimi, makine öğrenmesi, finansal analiz, sosyal bilimler ve daha birçok alanda veriyle çalışan herkes için vazgeçilmez bir araç haline gelmiştir. Pandas, özellikle yapısal (tablosal) ve zaman serisi verilerini kolayca işlemek ve analiz etmek için tasarlanmıştır.

Pandas'ın Temel Veri Yapıları: Pandas, veri manipülasyonu ve analizi için iki temel ve güçlü veri yapısı sunar:

Seriler (Series): Tek boyutlu, etiketlenmiş bir dizidir. NumPy dizilerine benzer ancak verilere ek olarak bir de indeks (etiket) bilgisi taşır. Bu indeks, verilere daha anlamlı bir şekilde erişmeyi ve onları hizalamayı sağlar. Bir Seri, farklı veri tiplerini (tam sayı, kayan nokta, metin vb.) içerebilir.

Veri Çerçeveleri (DataFrame): İki boyutlu, etiketlenmiş bir tablodur. Birden çok Serinin bir araya gelmesiyle oluşur ve sütunlar farklı veri tiplerini içerebilir. DataFrame'ler, SQL tablolarına veya Excel elektronik tablolarına benzer bir yapıya sahiptir ve veri analizi için en yaygın kullanılan Pandas veri yapısıdır. Satırlar ve sütunlar için ayrı indekslere sahiptir.

Pandas'ın Temel İşlevleri ve Yetenekleri: Pandas, veri manipülasyonu ve analizi süreçlerini kolaylaştıran geniş bir işlevsellik sunar:

- Veri Okuma ve Yazma: Farklı dosya formatlarından (CSV, Excel, SQL veritabanları, JSON, HTML vb.) veri okuma ve bu formatlara veri yazma imkanı sunar. Bu, farklı kaynaklardaki verilere kolayca erişmeyi ve sonuçları farklı formatlarda kaydetmeyi sağlar.
- Veri Temizleme: Eksik verileri (NaN - Not a Number) tespit etme ve işleme (silme, doldurma), yinelenen satırları bulma ve kaldırma, veri tiplerini dönüştürme gibi veri temizleme işlemlerini kolaylaştırır. Gerçek dünya verilerinin genellikle dağınık olduğu düşünüldüğünde, bu özellikler veri analizinin önemli bir parçasıdır.
- Veri Seçme ve Filtreleme: Etiketlere (indeks ve sütun adları) veya konumlara göre veri seçme, belirli koşulları sağlayan satırları veya sütunları filtreleme imkanı sunar. Bu, analiz için ilgilenilen veri alt kümelerine kolayca odaklanmayı sağlar.
- Veri Birleştirme ve Şekillendirme: Farklı DataFrame'leri birleştirme (merge, join), veri çerçevelerini yeniden şekillendirme (pivot, stack, unstack), veri çerçevelerini gruplama (groupby) ve toplu işlemler (aggregate) yapma gibi gelişmiş veri manipülasyonu yetenekleri sunar. Bu, farklı veri kaynaklarını bir araya getirme ve veriyi analiz için uygun bir formata dönüştürme imkanı sağlar.
- Veri Analizi ve İstatistiksel İşlemler: Verilerin temel istatistiksel özelliklerini hesaplama (ortalama, medyan, standart sapma vb.), korelasyon ve kovaryans hesaplama, frekans analizi yapma gibi çeşitli istatistiksel analizler gerçekleştirmeye olanak tanır.
- Zaman Serisi Veri Analizi: Zaman damgalı verileri (time series data) işlemek için özel araçlar sunar. Tarih aralıklarını yönetme, zaman kaydırma, yeniden örnekleme (resampling), hareketli ortalamalar hesaplama gibi zaman serisi özgü analizler yapmayı kolaylaştırır.
- Veri Görselleştirme: Matplotlib ve Seaborn gibi görselleştirme kütüphaneleriyle entegre olarak verilerin grafiksel olarak sunulmasına olanak tanır. Bu, veri örüntülerini ve ilişkilerini görsel olarak keşfetmeyi kolaylaştırır.

- Performans: Büyük veri setleriyle bile verimli bir şekilde çalışacak şekilde optimize edilmiştir. NumPy üzerine inşa edilmiş olması, vektörel işlemlerin hızlı bir şekilde gerçekleştirilmesini sağlar.

Pandas'ın Avantajları:

- Kullanım Kolaylığı: Sezgisel ve kullanıcı dostu bir API sunar. Veri manipülasyonu ve analizi için yüksek seviyeli araçlar sağlayarak kod yazma sürecini hızlandırır ve kolaylaştırır.
- Güçlü Veri Yapıları: Seri ve DataFrame veri yapıları, yapısal verileri etkili bir şekilde temsil etmeyi ve işlemeyi sağlar. Etiketlenmiş indeksleme, verilere anlamlı bir şekilde erişmeyi kolaylaştırır.
- Geniş İşlevsellik: Veri okuma/yazma, temizleme, manipülasyon, analiz ve görselleştirme için kapsamlı bir araç seti sunar. Çoğu veri analizi ihtiyacını tek bir kütüphane içinde karşılar.
- Topluluk Desteği: Büyük ve aktif bir topluluk tarafından desteklenmektedir. Bu, bol miktarda dokümantasyon, eğitim materyali ve yardım kaynağı anlamına gelir.
- Diğer Python Kütüphaneleriyle Entegrasyon: NumPy, Matplotlib, Seaborn, Scikit-learn gibi diğer popüler Python kütüphaneleriyle sorunsuz bir şekilde entegre olur. Bu, kapsamlı bir veri bilimi iş akışı oluşturmayı kolaylaştırır.
- Performans: C tabanlı optimizasyonlar sayesinde büyük veri setleriyle bile tatmin edici bir performans sunar.

Pandas'ın Dezavantajları:

- Bellek Tüketimi: Özellikle çok büyük veri setleriyle çalışırken bellek tüketimi sorun yaratabilir. Verinin tamamını belleğe yüklemesi gerekebilir. Ancak, büyük veri işleme için chunking gibi teknikler mevcuttur.
- Öğrenme Eğrisi (Başlangıçta): Geniş işlevselliği nedeniyle başlangıçta tüm özellikleri öğrenmek zaman alabilir. Ancak, temel kavramları öğrenmek oldukça kolaydır.

- Tek İşlemci Sınırlamaları (Bazı Durumlarda): Bazı işlemler tek bir işlemci çekirdeği üzerinde çalışabilir, bu da çok büyük veri setlerinde performansı sınırlayabilir. Ancak, paralelleştirme için Dask gibi kütüphanelerle entegre edilebilir.

6.2. Numpy

NumPy (Numerical Python), Python programlama dili için temel bir kütüphanedir ve özellikle büyük, çok boyutlu diziler (arrays) ve matrisler üzerinde etkili bir şekilde çalışmak üzere tasarlanmıştır. Travis Oliphant tarafından geliştirilmeye başlanan NumPy, bilimsel hesaplama, veri analizi, makine öğrenmesi ve mühendislik gibi birçok alanda Python ekosisteminin temelini oluşturur. Performans odaklı yapısı ve zengin matematiksel fonksiyonellikleri sayesinde, Python'ı güçlü bir sayısal hesaplama platformuna dönüştürmüştür.

NumPy'ın Temel Veri Yapısı: NumPy'ın kalbinde, ndarray (n-dimensional array) olarak adlandırılan çok boyutlu dizi nesnesi bulunur. Bu diziler, aynı türden (genellikle sayısal) verilerin homojen bir şekilde saklandığı ve üzerinde hızlı işlemlerin gerçekleştirilebildiği yapılardır. NumPy dizileri, Python listelerine benzer görünse de, performans ve işlevsellik açısından önemli farklılıklar sunar:

- Homojen Veri Tipi: Bir NumPy dizisindeki tüm elemanlar aynı veri tipinde olmalıdır (örneğin, tümü tam sayı veya tümü kayan nokta). Bu, bellek yönetimini optimize eder ve matematiksel işlemlerin daha hızlı yürütülmesini sağlar.
- Sabit Boyut: NumPy dizilerinin boyutu oluşturulduktan sonra genellikle sabittir (yeniden boyutlandırma mümkündür ancak maliyetli olabilir). Bu da bellek düzenlemesini kolaylaştırır.
- Vektörel Operasyonlar: NumPy, diziler üzerinde döngülere ihtiyaç duymadan toplu işlemler (vektörel operasyonlar) yapılmasına olanak tanır. Bu, geleneksel Python döngülerine kıyasla performansı önemli ölçüde artırır.

NumPy'ın Temel İşlevleri ve Yetenekleri: NumPy, bilimsel hesaplamaları kolaylaştıran geniş bir işlevsellik sunar.

- Dizi Oluşturma: Farklı şekillerde ve veri tiplerinde diziler oluşturmak için çeşitli fonksiyonlar sunar (örneğin np.array(), np.zeros(), np.ones(), np.arange(), np.linspace(), np.random.rand()).
- Dizi İndeksleme ve Dilimleme (Slicing): Dizilerin belirli elemanlarına veya alt kümelerine erişmek ve bunları değiştirmek için güçlü indeksleme ve dilimleme özellikleri sunar. Çok boyutlu dizilerde de esnek indeksleme imkanı sağlar.
- Temel Matematiksel Operasyonlar**:** Diziler arasında eleman bazında toplama, çıkarma, çarpma, bölme gibi temel aritmetik işlemleri hızlı bir şekilde gerçekleştirebilir. Ayrıca, skaler değerlerle dizi operasyonları da desteklenir (broadcasting).
- Evrensel Fonksiyonlar (ufuncs): Dizilerin tüm elemanlarına aynı anda uygulanan matematiksel fonksiyonlardır (örneğin np.sin(), np.cos(), np.exp(), np.log(), np.sqrt()). Bu fonksiyonlar, performansı optimize edilmiş C kodunda implemente edilmiştir.
- Lineer Cebir İşlemleri: Matris çarpımı (np.dot()), transpoz alma (.T), tersini alma (np.linalg.inv()), özdeğer ve özvektör hesaplama (np.linalg.eig()), denklem sistemlerini çözme (np.linalg.solve()) gibi temel lineer cebir işlemlerini destekler.
- İstatistiksel Fonksiyonlar: Dizilerin ortalamasını (np.mean()), medyanını (np.median()), standart sapmasını (np.std()), varyansını (np.var()), maksimum ve minimum değerlerini (np.max(), np.min()) bulma gibi çeşitli istatistiksel analizler için fonksiyonlar sunar.
- Şekil Manipülasyonu: Dizilerin şeklini değiştirme (np.reshape()), boyutlarını yeniden düzenleme (np.transpose()), dizileri birleştirme (np.concatenate(), np.stack()) ve bölme (np.split()) gibi dizi manipülasyon işlemlerini kolaylaştırır.
- Rastgele Sayı Üretimi: Farklı olasılık dağılımlarından rastgele sayılar ve diziler üretmek için fonksiyonlar içerir (np.random). Bu, simülasyonlar, istatistiksel örnekleme ve makine öğrenmesi uygulamaları için önemlidir.
- Dosya Girdisi/Çıktısı: Dizileri diske kaydetme (np.save(), np.savez()) ve diskten yükleme (np.load()) imkanı sunar.

NumPy'ın Avantajları:

- Performans: Vektörel operasyonlar ve optimize edilmiş C kodu sayesinde, geleneksel Python döngülerine kıyasla çok daha hızlı hesaplamalar gerçekleştirir. Bu, büyük veri setleriyle çalışırken kritik bir avantajdır.
- Bellek Verimliliği: Homojen veri yapısı sayesinde, Python listelerine göre daha az bellek kullanır.
- Geniş İşlevsellik: Bilimsel hesaplamalar için ihtiyaç duyulan temel matematiksel, lineer cebir ve istatistiksel fonksiyonların geniş bir yelpazesini sunar.
- Diğer Kütüphanelerle Entegrasyon: Pandas, SciPy, Matplotlib, Scikit-learn gibi diğer önemli bilimsel Python kütüphanelerinin temelini oluşturur ve onlarla sorunsuz bir şekilde entegre olur.
- Topluluk Desteği: Büyük ve aktif bir geliştirici topluluğu tarafından desteklenmektedir. Bu, kapsamlı dokümantasyon ve yardım kaynakları anlamına gelir.
- Çok Boyutlu Dizi Desteği: Yüksek boyutlu verileri (örneğin, görüntüler, tensörler) etkili bir şekilde temsil etme ve işleme imkanı sunar.

NumPy'ın Dezavantajları:

- Öğrenme Eğrisi (Bazı Konularda): Temel kavramlar kolay olsa da, bazı ileri düzey fonksiyonlar ve vektörel düşünce yapısı başlangıçta biraz zaman alabilir.
- Heterojen Veri Desteği Sınırlı: Dizilerdeki tüm elemanların aynı veri tipinde olması gerektiği için, farklı veri tiplerini bir arada tutmak gerektiğinde Pandas gibi daha esnek veri yapıları daha uygun olabilir.
- Yüksek Seviyeli Veri Analizi İçin Ek Kütüphanelere İhtiyaç: NumPy, temel sayısal hesaplamalar için güçlü olsa da, daha yüksek seviyeli veri manipülasyonu ve analizi (örneğin, eksik veri işleme, etiketlenmiş indeksleme) için genellikle Pandas gibi kütüphanelerle birlikte kullanılır.

6.3. NLTK (Natural Language Toolkit)

NLTK, Python programlama dili için geliştirilmiş, doğal dil işleme (NLP) görevlerini kolaylaştırmak amacıyla oluşturulmuş kapsamlı bir kütüphanedir. Steven Bird ve Edward Loper tarafından geliştirilmeye başlanan NLTK, akademik araştırmalar, eğitim ve endüstriyel uygulamalar için yaygın olarak kullanılan açık kaynaklı bir araçtır. NLTK, metin verilerini analiz etmek, anlamak ve işlemek için zengin bir araç seti, veri kaynakları (korpuslar) ve sözlükler sunar.

NLTK'nın Temel Amaçları ve Özellikleri:

- Kapsamlı Araç Seti: NLTK, metin işleme için temelden ileri seviyeye kadar birçok farklı görevi gerçekleştirebilecek modüller ve fonksiyonlar içerir. Bunlar arasında tokenleştirme, kök bulma (stemming), lemmatizasyon, etiketleme (part-of-speech tagging), ayrıştırma (parsing), anlambilimsel analiz ve daha fazlası bulunur.
- Eğitim ve Araştırma Odaklı: NLTK, NLP kavramlarını öğretmek ve araştırma yapmak için tasarlanmıştır. Kullanıcı dostu arayüzü ve kapsamlı dokümantasyonu sayesinde NLP alanına yeni başlayanlar için ideal bir öğrenme aracıdır. Aynı zamanda, ileri düzey araştırmacılar için de güçlü analiz yetenekleri sunar.
- Zengin Veri Kaynakları (Corpora) ve Sözlükler: NLTK, çeşitli dillerde büyük metin koleksiyonları (korpuslar) ve sözlükler (örneğin WordNet) içerir. Bu kaynaklar, NLP algoritmalarını eğitmek, test etmek ve dilbilgisel bilgileri kullanmak için değerli veriler sağlar.
- Geniş Topluluk Desteği: NLTK, aktif bir geliştirici ve kullanıcı topluluğuna sahiptir. Bu, sorulara hızlı yanıt bulma, yeni araçlar ve kaynaklar keşfetme ve kütüphanenin gelişimine katkıda bulunma imkanı sunar.
- Platform Bağımsızlık: Python üzerinde çalıştığı için, NLTK de Windows, macOS ve Linux gibi farklı işletim sistemlerinde sorunsuz bir şekilde kullanılabilir.

- Modüler Tasarım: NLTK, farklı NLP görevleri için ayrı modüller sunar. Bu modüler yapı, kullanıcıların yalnızca ihtiyaç duydukları araçları içe aktarmasına ve kullanmasına olanak tanır, bu da kodun daha düzenli ve verimli olmasını sağlar.

NLTK'nın Avantajları:

- Kapsamlılık: Çok çeşitli NLP görevleri için geniş bir araç yelpazesi sunar.
- Eğitim Odaklılık: Öğrenmesi kolay yapısı ve kapsamlı dokümantasyonu sayesinde NLP öğrenmek için idealdir.
- Zengin Kaynaklar: Çok sayıda korpus ve sözlük içerir.
- Topluluk Desteği: Aktif ve yardımsever bir topluluğa sahiptir.
- Modülerlik: İhtiyaç duyulan araçların kolayca içe aktarılmasını sağlar.
- Platform Bağımsızlık: Python'ın desteklediği her platformda çalışır.

NLTK'nın Dezavantajları:

- Performans (Bazı Durumlarda): Büyük veri setleriyle çalışırken bazı NLTK fonksiyonları, daha optimize edilmiş kütüphanelere (örneğin spaCy) kıyasla daha yavaş olabilir.
- Bazı Algoritmaların Temel Seviyede Olması: Bazı temel NLP algoritmalarını içerse de, en son ve en gelişmiş derin öğrenme tabanlı NLP modellerini doğrudan sunmaz (bu tür modeller için genellikle TensorFlow, PyTorch gibi kütüphaneler kullanılır).
- Veri Ön İşleme Gereksinimi: NLTK araçlarının etkili bir şekilde çalışabilmesi için metin verisinin uygun şekilde ön işlenmesi (temizlenmesi, tokenleştirilmesi vb.) önemlidir.

6.4. Scikit-Learn (SKLEARN)

Scikit-learn, Python programlama dili için geliştirilmiş, açık kaynaklı ve ticari amaç gütmeyen bir makine öğrenmesi kütüphanesidir. Basit ve tutarlı bir API (Application Programming Interface) sunarak sınıflandırma, regresyon, kümeleme, boyut indirgeme, model seçimi, veri ön işleme ve daha birçok makine öğrenmesi görevini kolaylaştırmayı amaçlar. NumPy, SciPy ve Matplotlib gibi diğer bilimsel

Python kütüphaneleri üzerine inşa edilmiştir ve veri bilimi ve makine öğrenmesi alanlarında Python ekosisteminin temel taşlarından biridir.

Scikit-learn'ün Temel Amaçları ve Özellikleri:

- Kapsamlı Algoritma Yelpazesi: Scikit-learn, denetimli öğrenme (sınıflandırma, regresyon), denetimsiz öğrenme (kümeleme, boyut indirgeme), model seçimi (hiperparametre ayarlama, model değerlendirme) ve veri ön işleme için çok çeşitli popüler ve etkili algoritmalar sunar.
- Tutarlı ve Kullanıcı Dostu API: Kütüphanedeki tüm algoritmalar ve araçlar, öğrenmeyi ve kullanmayı kolaylaştıran tutarlı bir yapıya sahiptir. Temel adımlar genellikle modelin oluşturulması (estimator), veriye uyarlanması (fit), tahmin yapılması (predict, predict_proba) ve performansın değerlendirilmesi (score, metrik fonksiyonları) şeklindedir.
- Kolay Entegrasyon: NumPy ve Pandas gibi diğer bilimsel Python kütüphaneleriyle sorunsuz bir şekilde entegre olur. Veri manipülasyonu ve sayısal işlemler için bu kütüphanelerin güçlü özelliklerinden yararlanır.
- Kapsamlı Dokümantasyon ve Topluluk Desteği: Scikit-learn, anlaşılır ve detaylı bir kullanıcı kılavuzuna ve API referansına sahiptir. Ayrıca, aktif bir geliştirici ve kullanıcı topluluğu sayesinde sorulara hızlı yanıt bulmak ve bilgi alışverişinde bulunmak kolaydır.
- Performans Odaklı: Algoritmaların çoğu, performansı optimize edilmiş C veya Cython dillerinde implemente edilmiştir. Bu, büyük veri setleriyle bile verimli bir şekilde çalışmayı sağlar.
- Model Seçimi ve Değerlendirme Araçları: Modelin performansını değerlendirmek için çeşitli metrikler (doğruluk, kesinlik, recall, F1-skoru, RMSE vb.) ve model seçimi teknikleri (çapraz doğrulama, grid arama) sunar. Bu, en uygun modeli seçmeyi ve hiperparametrelerini ayarlamayı kolaylaştırır.
- Veri Ön İşleme Araçları: Veri ölçeklendirme, normalleştirme, özellik seçimi, boyut indirgeme gibi makine öğrenmesi modelleri için önemli olan veri ön işleme adımlarını gerçekleştirmek için çeşitli araçlar sunar.
- Pipeline (Boru Hattı) İmkanı: Veri ön işleme adımlarını ve modelleme adımlarını tek bir ardışık yapı (pipeline) içinde birleştirmeyi sağlar. Bu, kodun

daha düzenli, okunabilir ve tekrarlanabilir olmasını sağlar ve modelin tutarlılığını artırır.

Scikit-learn'ün Temel Modülleri ve İşlevleri:

Scikit-learn, farklı makine öğrenmesi görevlerini destekleyen çeşitli modüllere ayrılmıştır:

- sklearn.datasets (Veri Setleri): Öğrenme ve prototipleme için kullanılabilecek çeşitli yerleşik veri setleri (örneğin Iris, Boston Housing) ve veri oluşturma araçları içerir.
- sklearn.preprocessing (Veri Ön İşleme): Veri ölçeklendirme (StandardScaler, MinMaxScaler), normalleştirme (Normalizer), kategorik değişken kodlama (OneHotEncoder, LabelEncoder), eksik veri işleme (SimpleImputer), özellik seçimi (SelectKBest, RFE) gibi veri ön işleme tekniklerini içerir.
- sklearn.feature_extraction (Özellik Çıkarma): Metin ve görüntü gibi yapılandırılmamış verilerden sayısal özellikler çıkarmak için araçlar sunar (örneğin TfidfVectorizer, CountVectorizer metin için; FeatureHasher genel özellik çıkarma için).
- sklearn.feature_selection (Özellik Seçimi): Modelin performansı üzerinde en etkili olan özellikleri seçmek için çeşitli yöntemler sunar (örneğin SelectKBest, RFE).
- sklearn.decomposition (Boyut İndirgeme): Yüksek boyutlu veriyi daha düşük boyutlu bir temsile indirgemek için teknikler içerir (örneğin PCA, TruncatedSVD).
- sklearn.linear_model (Doğrusal Modeller): Doğrusal regresyon (LinearRegression), lojistik regresyon (LogisticRegression), destek vektör makinelerinin doğrusal versiyonları (LinearSVC, LinearSVR) gibi doğrusal modelleri içerir.
- sklearn.svm (Destek Vektör Makineleri - SVM): Sınıflandırma (SVC) ve regresyon (SVR) için güçlü SVM algoritmalarını sunar.
- sklearn.tree (Karar Ağaçları): Karar ağacı tabanlı sınıflandırma (DecisionTreeClassifier) ve regresyon (DecisionTreeRegressor) modellerini içerir.

- sklearn.ensemble (Ensemble Yöntemleri): Birden çok temel modeli birleştirerek daha güçlü tahminler yapan topluluk öğrenme yöntemlerini sunar (örneğin RandomForestClassifier, RandomForestRegressor, GradientBoostingClassifier, AdaBoostClassifier).
- sklearn.neighbors (En Yakın Komşu Algoritmaları): K-En Yakın Komşu (KNN) sınıflandırma (KNeighborsClassifier) ve regresyon (KNeighborsRegressor) algoritmalarını içerir.
- sklearn.cluster (Kümeleme): Veri noktalarını benzer özelliklere göre gruplandırmak için algoritmalar sunar (örneğin KMeans, AgglomerativeClustering, DBSCAN).
- sklearn.model_selection (Model Seçimi ve Değerlendirme): Veri bölme (train_test_split), çapraz doğrulama (cross_val_score, KFold), hiperparametre ayarlama (GridSearchCV, RandomizedSearchCV) ve performans metrikleri (accuracy_score, classification_report, mean_squared_error vb.) için araçlar içerir.
- sklearn.pipeline (Boru Hattı): Veri ön işleme ve modelleme adımlarını sıralı bir şekilde birleştirmek için Pipeline sınıfını sunar.
- sklearn.metrics (Metrikler): Sınıflandırma, regresyon ve kümeleme modellerinin performansını değerlendirmek için çeşitli metrik fonksiyonları içerir.

Scikit-learn'ün Avantajları:

- Kullanım Kolaylığı: Tutarlı API ve anlaşılır dokümantasyon sayesinde makine öğrenmesi algoritmalarını uygulamak ve denemek kolaydır.
- Geniş Algoritma Yelpazesi: Çeşitli denetimli, denetimsiz ve model seçimi algoritmaları sunar.
- Güçlü Temel: NumPy ve SciPy üzerine inşa edildiği için yüksek performanslı sayısal işlemlerden yararlanır.
- Kapsamlı Araçlar: Veri ön işleme, özellik mühendisliği, model seçimi ve değerlendirme için gerekli araçları içerir.
- Açık Kaynak ve Topluluk Desteği: Geniş ve aktif bir topluluk tarafından desteklenir ve sürekli olarak geliştirilmektedir.

- Endüstri Standardı: Makine öğrenmesi projelerinde yaygın olarak kullanılan bir kütüphanedir.

Scikit-learn'ün Dezavantajları:

- Derin Öğrenme Desteği Sınırlı: Temel makine öğrenmesi algoritmalarına odaklanır ve derin öğrenme modelleri için TensorFlow veya PyTorch gibi özel kütüphaneler daha uygundur. Ancak, Scikit-learn ile temel derin öğrenme modelleri oluşturmak mümkündür.
- Büyük Veri Setleri İçin Ölçeklenebilirlik: Çok büyük veri setleriyle çalışırken bazı algoritmaların ölçeklenebilirliği sınırlı olabilir. Bu tür durumlar için Spark MLlib gibi daha büyük veri işleme çerçeveleri gerekebilir.
- Özelleştirme Zorluğu (Bazı Durumlarda): Yüksek seviyeli bir API sunduğu için, bazı algoritmaların iç mekanizmalarına derinlemesine müdahale etmek veya tamamen özel algoritmalar geliştirmek daha zor olabilir.

7. UYGULAMA

Bu çalışmada geliştirilen uygulamanın temel amacı, belirli bir veri seti üzerinde duygu analizi gerçekleştirerek, elde edilen analiz sonuçlarını çeşitli makine öğrenmesi algoritmalarına entegre edip tahminleme yapmaktır. Uygulamanın geliştirilme sürecinde Python programlama dili tercih edilmiş ve farklı teknolojilerin projeye entegre edilmesi için çeşitli kütüphanelerden faydalanılmıştır. Proje yapılırken Python programlama dili yanında Makine öğrenmesi teknolojisinin yöntemlerinden Denetimli Öğrenme yönteminin örneği olan Naive Bayes, Destek Vektör Makineleri(SVM) ve Random Forest algoritmaları, yine sınıflandırma yöntemleriyle birlikte kullanılan Feature Selection(Özellik Seçimi) yöntemlerinden Ki-Kare Testi (Chi-Squared Test) yöntemi ve Denetimsiz Öğrenme yöntemi algoritmaslarından da Kmeans ve PCA algoritmaları kullanılmıştır. Ayrıca bu yöntemler yardımıyla hibrit yaklaşımlar da modellenmiştir. Bu hibrit modellemelerle mevcut tekil modellenen yöntemlerden daha iyi sonuç elde edilmeye çalışılmıştır.

Yapılan uygulamanın ilk aşamasında Kaggle üzerinden elde edilen veri setindeki 929.544 verinin sadece 10.000 tanesi modele gönderilmiştir. Modele gönderilen verilerin 7.000 tanesi train, 3.000 tanesi test verisidir. Kaggle'dan alınan veri setinde farklı dillerde metin verileri mevcuttu. Bu çalışmada İngilizce veriler kullanılacağı için veri setinden sadece İngilizce veriler seçilmiş ve modele gönderilmiştir. Kaggle'dan alınan veri setinde bulunan 929.544 verinin %28'i pozitif, %28'i negatif, kalan %44'lük kısmı da farklı duygu etiketlerine sahiptir. Bu çalışmada sadece pozitif ve negatif duygu etiketli veriler kullanılmıştır. Modele verilen pozitif veya negatif duygu etiketi içeren veriler, Kaggle'dan alınan veri seti içerisinden tamamen rastgele seçilmiştir fakat bu seçim esnasında verilerin homojen dağılımına dikkat edilmiştir. Train veri setindeki pozitif etiket içeren veri sayısı 3451 yani veri setinin %49,3'ü, negatif etiket içeren veri sayısı 3549 yani veri setinin %50,7'sidir. Test veri setindeki pozitif etiket içeren veri sayısı 1475 yani veri setinin %49,17'si, negatif etiket içeren veri sayısı 1525 yani veri setinin %50,83'üdür. Excel dosyalarının ilk kolonundaki veri Twitter metin verisi, ikinci kolondaki veri ise ilk kolondaki metin verisinin sahip olduğu duyguyu gösteren etiket değeridir. Modelleme bu varsayıma göre yapılmıştır.

Mevcut veri setinin etiket değerleri olarak pozitif ve negatif etiket değerleri bulunmaktadır. Etiketli olan mevcut veri seti denetimli öğrenme algoritmaları ile modellenebilir durumdadır. Bu aşamada sınıflandırma yöntemlerinden Naive Bayes, SVM ve Random Forest algoritmaları seçilmiş ve modeller yapılmıştır. Ayrıca modellenen hibrit yaklaşımlar için de aynı test ve train veri setleri kullanılmıştır. Dolayısıyla tahminleme işlemleri sadece SVM, sadece Random Forest ve sadece Naive Bayes ile ayrı ayrı modellenmiş bunlara ek olarak da hibrit yaklaşımlar geliştirilmiştir. Hibrit yaklaşımlar modellenirken mevcut sınıflandırma yöntemlerinin doğru tahminleme oranlarının üzerine çıkarak daha iyi doğru tahminleme oranları elde edilmeye çalışılmıştır.

Uygulamanın gerçekleştirilmesi aşamasında ilk olarak önceden test ve train verisi olmak üzere belirlenen Excel dosyaları model üzerinde işlenmek üzere kaydedilmiştir. Modelleme 10.000 veri üzerinde yapılmıştır. Bu verilerin 7.000 tanesi modelin eğitilmesi için kullanılmış, geriye kalan 3.000 veri ise test aşamasında modelin doğruluğunu ölçmek amacıyla ayrılmıştır. Kullanılan veri seti, işleme alınmadan önce sadeleştirilmiş ve analiz sonucunu doğrudan etkilemeyen bazı sütunlar çıkarılmıştır.

Veri temizleme aşamasında, her bir tweet üzerinde çeşitli ön işlemler gerçekleştirilmiştir. Öncelikli olarak URL'ler, Twitter kullanıcı adları, hashtag'ler, özel karakterler, emojiler, sayılar, noktalama işaretleri, harf tekrarları ve fazla boşluklar temizlenmiştir. Temizleme işlemi tamamlandıktan sonra, veriler kelimelerine ayrılmış, ardından kelimeler köklerine indirgenmiştir. Bu süreçlerin Python programlama dili ile gerçekleştirilmesi için NLTK kütüphanesi projeye entegre edilmiştir. NLTK, doğal dil işleme (NLP) kütüphanesidir ve Python tabanlıdır.

Metin işleme aşamasında, temizlenmiş veriler pipeline adı verilen bir işleme hattına dahil edilmiş ve belirlenen adımlar doğrultusunda analiz edilmiştir. Analizin bir parçası olarak, stop words (and, or, the, is, to, for vb.) olarak bilinen gereksiz kelimeler metinlerden çıkarılmıştır. Bu işlemin tamamlanmasının ardından, veriler duygu analizi için uygun hale getirilmiştir.

Yazılan modelde makine öğrenmesi algoritmalarını projeye entegre edebilmek için Scikit-Learn kütüphanesi kullanılmıştır. Scikit-Learn, makine öğrenmesi ve veri bilimi alanlarında Python ekosisteminin, en önemli açık kaynaklı yazılım kütüphanelerinden biridir.

Önceden Kaggle üzerinden elde edilen 10.000 verinin 7.000 tanesi modelin eğitilmesi amacıyla kullanılmıştır. Yapılan duygu analizi sonucunda bu 7.000 veri başarılı bir şekilde işlenmiş ve modelin eğitimi tamamlanmıştır. Modelin doğruluğunu test edebilmek için daha önce ayrılmış olan 3.000 adet test verisi modele verilmiştir. Bu aşamada, test verilerinin gerçek duygu etiketlerinin bilinmesi gerekmektedir. Gerçek duygu etiketleri bilinmediği takdirde, makine öğrenmesi algoritmalarının tahminlerinin doğruluk oranı tespit edilemeyecektir.

Modele gönderilen verilerin tamamının metin verisi olduğu göz önünde bulundurulduğunda bu aşamada bir metin madenciliği yöntemi kullanılmalıdır. Metin madenciliği ile, önceki aşamalarda modele gönderilen veri setinde bulunan cümlelelerin her bir kelimesi için ağırlık hesabı yapılmalıdır. Bu aşamada metin madenciliğinde ve doğal dil işleme (NLP) uygulamalarında sıkça kullanılan bir özellik çıkarım (feature extraction) yöntemi olan TF-IDF (Term Frequency - Inverse Document Frequency) yöntemi kullanılmıştır. Bu yöntemi kullanmaktaki amaç, metinlerdeki önemli kelimeleri belirlemek ve bunları sayısal vektörlere dönüştürmektir. Sayısal vektörlere dönüştürülen veriler, n-gram adı verilen bir dil modellemesi yöntemi yardımıyla metni gruplara ayırarak dedğerlendirilmiştir. Bu yapı, metni kelime gruplarına veya karakter gruplarına bölerek dilin yapısını sayısal olarak temsil etmeye yarar. Bu çalışmada unigram, bigram ve trigram yapılarıyla ayrı ayrı denemeler yapılmış ve modeller özelinde hangi yapı en iyi sonucu vermişse o modelde o yapı kullanılmıştır.

Son aşamada, makine öğrenmesi algoritmalarının eğitilmesi ve test edilmesi işlemleri gerçekleştirilmiştir. Train verileri makine öğrenmesi modeline öğretilmiş ve test verilerinin tahmin edilmesi sağlanmıştır. Çalışmada üç farklı makine öğrenmesi sınıflandırma algoritması kullanılmıştır: Support Vector Machine (SVM), Random Forest (RF) ve Naive Bayes (NB) yöntemleri tekil olarak modellenmiştir.

Bunun yanında hibrit yaklaşımlar da farklı varyasyonlarla modellenmiştir. Tekil modellerde en iyi sonucu SVM algoritması verdiği için hibrit yöntemlerde de ilk aşamada sınıflandırıcı olarak SVM kullanılmıştır. Sonrasında ise çalışmayı daha da derinleştirmek amacıyla SVM yerine Random Forest yöntemi kullanılarak da hibrit yaklaşımlar çeşitlendirilmiştir.

Bütün yaklaşımlar modellenirken kullanılan algoritmaların en iyi sonuç verdiği parametreler tespit edilerek modellenilmiştir. Bu tespit yapılırken GridSearchCV kullanılmıştır. En iyi parametreler belirlenmiş, daha sonra belirlenen parametreler modellere sabit değer olarak verilerek modellenmiştir. Modellerde kullanılan sınıflandırma yöntemlerinden SVM için kullanılan parametreler şunlardır:
C değeri: Hatalı sınıflandırmalara karşı modelin ne kadar tolerans göstereceğini belirler. Bu değer modele verilirken 3 farklı değer ile test yapılmıştır. 0.1, 1 ve 10 olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Kernel değeri: Veriyi yüksek boyutlu uzaya dönüştürerek doğrusal olmayan verileri ayırmaya olanak tanır. Bu değer modele verilirken 2 farklı değer ile test yapılmıştır. Linear ve rbf olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Gamma değeri: Tekil verilerin model üzerindeki etkisini belirler. Bu değer modele verilirken 3 farklı değer ile test yapılmıştır. 0.1, 1 ve scale olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Modellerde kullanılan sınıflandırma yöntemlerinden Random Forest için kullanılan parametreler şunlardır:

n_estimators değeri: Karar ağaçlarının sayısını belirtir. Bu değer modele verilirken 2 farklı değer ile test yapılmıştır. 100 ve 200 olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

max_depth değeri: Karar ağaçlarının derinliğini belirtir. Yani her ağacın en kadar dallanacağını gösterir. Burada sadece none değeri seçilmiştir yani model ne kadar dallanacağına kendi karar vermiştir.

min_samples_split değeri: Bir düğümün daha fazla bölünebilmesi için gereken minimum örnek sayısıdır. Burada 2 farklı değer ile test yapılmıştır. 2 ve 5 olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Naive Bayes için veriler metin verisi olduğundan ve vektörizasyon için TF-IDF kullanılmasından dolayı MultinominalNB yöntemi kullanılmıştır. Modellerde kullanılan sınıflandırma yöntemlerinden Naive Bayes için kullanılan parametreler şunlardır:

alpha değeri: Eğitim verisinde hiç görülmeyen kelimeler ya da özellikler test verisinde geldiğinde, Naive Bayes çarpma işlemi kullandığı için sıfır olasılık üretir. Bu da tüm hesaplamayı sıfırlayıp kötü tahminlere sebep olur. Bu durumu önlemek içinalpha değeri kullanılır. Burada 2 farklı değer ile test yapılmıştır. 0.1 ve 1.0 olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Modellerde kullanılan kümeleme yöntemlerinden PCA için kullanılan parametreler şunlardır:

n_components değeri: Kaç adet ana bileşen (principal component) alınacağını belirler. Burada tek değer ile test yapılmıştır. Literatürde yaygın olarak kullanılan değer ve veri sayısı göz önüne alındığında bu değer seçilmiştir. 300 olarak modele gönderilmiştir.

Modellerde kullanılan kümeleme yöntemlerinden Kmeans için kullanılan parametreler şunlardır:

n_clusters değeri: Kaç adet ana bileşen (principal component) alınacağını belirler. Burada 2 farklı değer ile test yapılmıştır. 3 ve 5 olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Modellerde kullanılan feature selection yöntemlerinden Ki-Kare Testi için kullanılan parametreler şunlardır:

score_func değeri: Her özelliğin (örneğin bir kelimenin), hedef değişken (etiket) ile istatistiksel olarak ne kadar ilişkili olduğunu hesaplar. Burada yöntem Ki-Kare Testi yöntemi kullanılacağı için modele chi2 değeri verilmiştir.

K Değeri: En yüksek chi2 skoruna sahip özellikleri (kelimeleri) seçer. Geri kalanları atar boyut indirimi yapılmış olur. 1000 ve 2000 olarak modele gönderilmiş en iyi sonucun seçilmesi beklenmiştir.

Bu çalışmada vektörizasyon yöntemi olarak sadece TF-IDF kullanılmıştır. Çalışmada kullanılan tüm model varyasyonları aşağıdaki gibidir ve toplam 9 tanedir. Tüm modellerde GridSearchCV yöntemi ile en iyi doğru tahminleme sonucu veren parametreler belirlenmiştir ve sadece seçilen parametrelerle modellemeler yapılmıştır:

- Sadece SVM kullanılarak yapılan tahminlemede metinler bigram ile gruplandırılmış, SVM parametreleri ise ceza parametresi olarak 1, kernel seçimi olarak linear, gamma parametresi scale olarak belirlenmiştir.
- Sadece Random Forest kullanarak yapılan tahminlemede metinler unigram ile gruplandırılmış, karar ağacı sayısı 200, karar ağacının derinliği sınırsız bırakılmış ve bir düğümün bölünen minimum örnek sayısı 2 olarak belirlenmiştir.
- Sadece Naive Bayes kullanılarak yapılan tahminlemede metinler bigram ile gruplandırılmış, alpha değeri 1.0 olarak belirlenmiştir.
- Kümeleme yöntemi olarak PCA, sınıflandırıcı olarak SVM kullanarak yapılan tahminlemede metinler trigram ile gruplandırılmış, PCA ile orijinal özellik boyutu 300'e düşürülmüş, SVM parametreleri ise ceza parametresi 10, kernel parametresi rbf, gamma parametresi 1 olarak belirlenmiştir.
- Kümeleme yöntemi olarak PCA, sınıflandırıcı olarak Random Forest kullanarak yapılan tahminlemede metinler bigram ile gruplandırılmış, PCA ile orijinal özellik boyutu 300'e düşürülmüş, karar ağacı sayısı 200, karar ağacının derinliği sınırsız bırakılmış ve bir düğümün bölünen minimum örnek sayısı 2 olarak belirlenmiştir.
- Kümeleme yöntemi olarak Kmeans, sınıflandırıcı olarak SVM kullanarak yapılan tahminlemede metinler bigram ile gruplandırılmış, küme sayısı 5 olarak verilmiş, SVM parametreleri ise ceza parametresi 10, kernel parametresi rbf, gamma parametresi scale olarak belirlenmiştir.

- Kümeleme yöntemi olarak Kmeans, sınıflandırıcı olarak Random Forest kullanarak yapılan tahminlemede metinler unigram ile gruplandırılmış, küme sayısı 5 olarak verilmiş, karar ağacı sayısı 100, karar ağacının derinliği sınırsız bırakılmış ve bir düğümün bölünen minimum örnek sayısı 2 olarak belirlenmiştir.
- Feature Selection yöntemi olarak Ki-Kare testi, sınıflandırıcı olarak SVM kullanılarak yapılan tahminlemede metinler bigram ile gruplandırılmış, Ki-Kare testi ile en anlamlı yani en yüksek skora sahip 1000 kelime seçilmiş, SVM parametreleri ise ceza parametresi 1, kernel parametresi rbf, gamma parametresi 1 olarak belirlenmiştir.
- Feature Selection yöntemi olarak Ki-Kare testi, sınıflandırıcı olarak Random Forest kullanılarak yapılan tahminlemede metinler bigram ile gruplandırılmış, Ki-Kare testi ile en anlamlı yani en yüksek skora sahip 1000 kelime seçilmiş, karar ağacı sayısı 100, karar ağacının derinliği sınırsız bırakılmış ve bir düğümün bölünen minimum örnek sayısı 2 olarak belirlenmiştir.

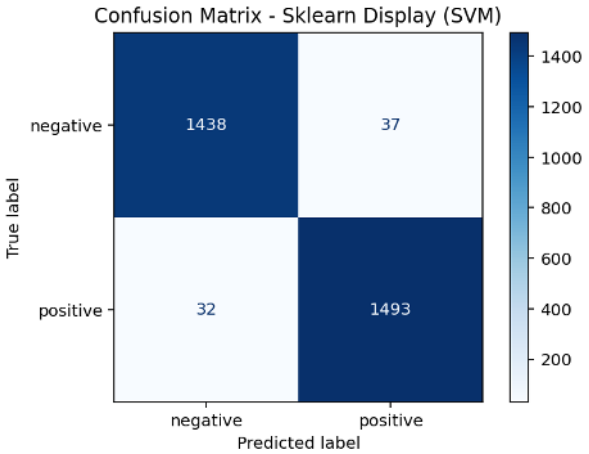
Sonuç olarak geliştirilen uygulama, veri setleri üzerinde duygu analizi gerçekleştirerek, makine öğrenmesi algoritmaları yardımıyla tahminleme yapabilen birden fazla model ortaya koymuştur. Kullanılan farklı makine öğrenmesi algoritmaları sayesinde 9 farklı model oluşturulmuş, bu modellerin doğruluk oranları analiz edilmiş ve sonuçlar değerlendirilmiştir. Hibrit yaklaşımlar ile sadece sınıflandırıcı kullanılan yöntemlere göre daha iyi tahminleme oranları elde edilmiştir. Bu çalışma, duygu analizi ve makine öğrenmesi alanında gerçekleştirilen başarılı bir uygulama olarak, gelecekte yapılacak benzer çalışmalar için önemli bir referans niteliği taşımaktadır.

SONUÇ

Bu çalışmada, metin verilerinden duygu analizi ve sınıflandırma yapmak amacıyla geliştirilen sistemin performansı, çeşitli makine öğrenimi modelleri kullanılarak değerlendirilmiştir. Eğitim verileri üzerinde eğitilen modeller, ayrı bir test veri seti üzerinde test edilerek doğruluk oranları hesaplanmıştır. Elde edilen sonuçlar, uygulanan metodolojinin etkinliğini ve model performanslarını karşılaştırmalı olarak sunmaktadır.

Yapılan değerlendirme sonucunda, dokuz farklı modelin performansı karşılaştırılmıştır: Support Vector Machine (SVM), Naive Bayes (Multinominal NB), Random Forest, PCA + SVM, PCA + Random Forest, Ki-Kare + SVM, Ki-Kare + Random Forest, Kmeans + SVM ve Kmeans + Random Forest. Her bir modelin doğruluk oranı, test veri seti üzerinde yapılan tahminlerin gerçek etiketlerle karşılaştırılmasıyla hesaplanmıştır.

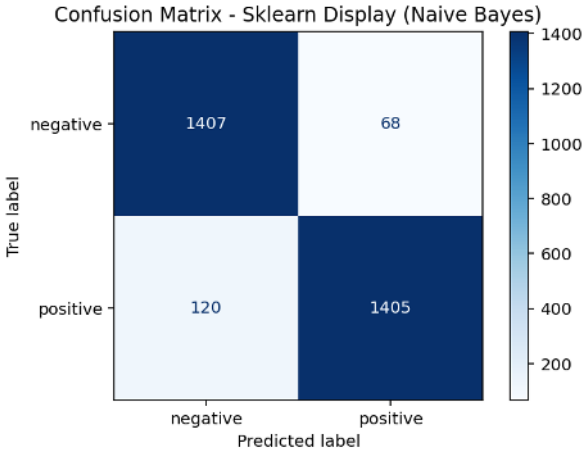
Destek Vektör Makineleri (SVM) modeli, test verileri üzerinde kaydettiği %97.70'lik doğru tahminleme oranıyla, tek başına sınıflandırıcı kullanılan yöntemler arasında en yüksek başarıyı göstermiştir. Bu sonuç, SVM algoritmasının metin sınıflandırma görevlerinde ne kadar etkili olduğunu açıkça ortaya koymaktadır. Özellikle SVM'in yüksek boyutlu veri uzaylarında dahi üstün performans sergileyebilme kabiliyeti, metin verilerinin barındırdığı karmaşık ilişkileri başarıyla modellemesini sağlamıştır.



Şekil 12. SVM Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

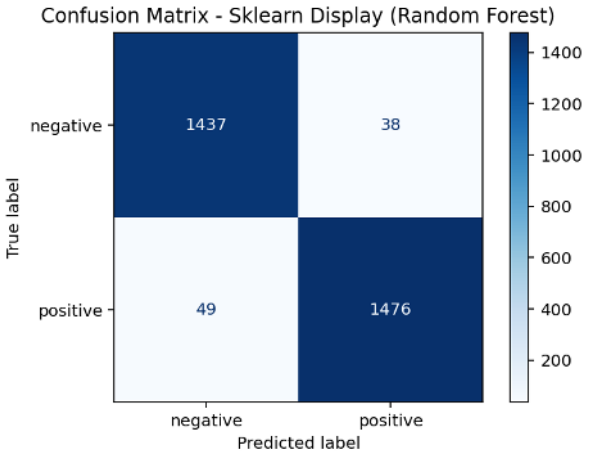
Öte yandan, Naive Bayes modeli test verileri üzerinde %93.73'lük bir doğru tahminleme oranıyla, diğer iki yönteme kıyasla daha mütevazı bir performans sergilemiştir. Bu durum, Naive Bayes algoritmasının metin verilerindeki karmaşık ilişkileri ve kelimeler arasındaki potansiyel bağımlılıkları yeterince modelleyememesinden kaynaklanabilir. Temelinde güçlü bir bağımsızlık varsayımına dayanan Naive Bayes, metinlerdeki kelimelerin birbirleriyle olan etkileşimini göz ardı edebilmektedir.



Şekil 13. Naive Bayes Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

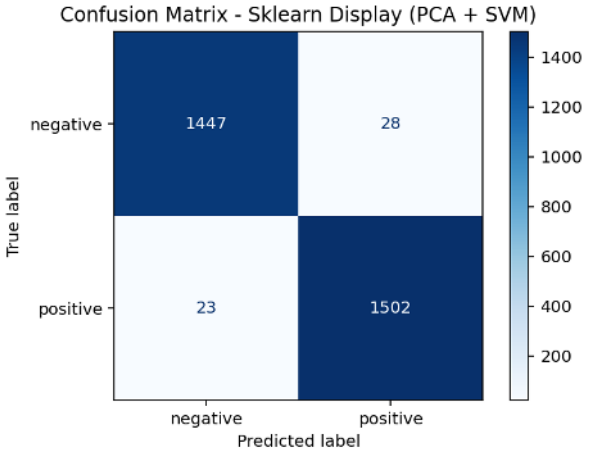
TF-IDF ve Random Forest modelinin birlikte kullanımı, test verileri üzerinde %97.10'luk bir doğru tahminleme oranı elde ederek, metin sınıflandırma görevinde yüksek bir başarı sergilemiştir. Bu sonuç, TF-IDF ile metin verilerinin anlamlı sayısal özelliklere dönüştürülmesinin ve Random Forest algoritmasının bu özellikler üzerinde etkili bir şekilde sınıflandırma yapabilmesinin bir göstergesidir. Özellikle, Random Forest'ın karar ağaçlarından oluşan yapısı sayesinde, metin verisindeki karmaşık ilişkileri başarıyla modelleyebildiği görülmektedir. Elde edilen yüksek doğruluk oranı, bu yöntemin benzer metin sınıflandırma problemleri için umut vadeden bir yaklaşım olduğunu desteklemektedir.



Şekil 14. Random Forest Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

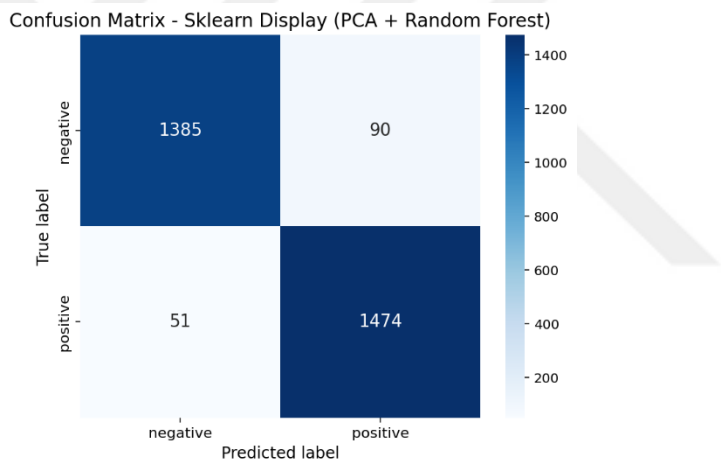
TF-IDF, PCA ve SVM algoritmalarının ardışık olarak uygulandığı bu model, test verileri üzerinde %98.30'luk etkileyici bir doğru tahminleme oranına ulaşmıştır. Bu sonuç, öncelikle TF-IDF yöntemiyle metin verilerinin zengin ve anlamlı özelliklere dönüştürülmesinin, ardından PCA ile bu özellik uzayının boyutunun önemli ölçüde azaltılarak bilgi kaybının minimize edilmesinin ve son olarak SVM algoritmasının bu optimize edilmiş özellikler üzerinde yüksek doğrulukla sınıflandırma yapabilmesinin birleşimi sayesinde elde edilmiştir. PCA'nın yüksek boyutlu metin verisindeki gürültüyü azaltma ve en önemli varyansı yakalama yeteneği, SVM'in daha etkili bir şekilde çalışmasını sağlamış olabilir. Elde edilen bu yüksek performans, bu üçlü yöntemin metin sınıflandırma problemlerinde güçlü bir alternatif oluşturduğunu göstermektedir.



Şekil 15. PCA + SVM Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

TF-IDF, PCA ve Random Forest algoritmalarının bir arada kullanıldığı bu model, test verileri üzerinde %95.30'luk bir doğruluk oranı elde etmiştir. Bu sonuç, metin verilerinin öncelikle TF-IDF ile kelime seviyesinden üçlü kelime gruplarına kadar (ngram_range = (1, 2)) geniş bir bağlamda sayısal özelliklere dönüştürülmesinin, ardından PCA ile bu yüksek boyutlu özellik uzayının 300 boyuta düşürülerek daha yönetilebilir hale getirilmesinin ve son olarak Random Forest algoritmasının bu indirgenmiş özellikler üzerinde etkili bir sınıflandırma performansı sergilemesinin bir sonucudur. PCA'nın boyut indirgeme yeteneği, Random Forest gibi ağaç tabanlı algoritmaların daha hızlı eğitilmesine ve potansiyel olarak aşırı öğrenmenin azaltılmasına katkıda bulunmuş olabilir. Elde edilen %95.30'luk doğruluk oranı, bu birleşik yaklaşımın metin sınıflandırma görevlerinde dikkate değer bir başarıya ulaştığını göstermektedir.

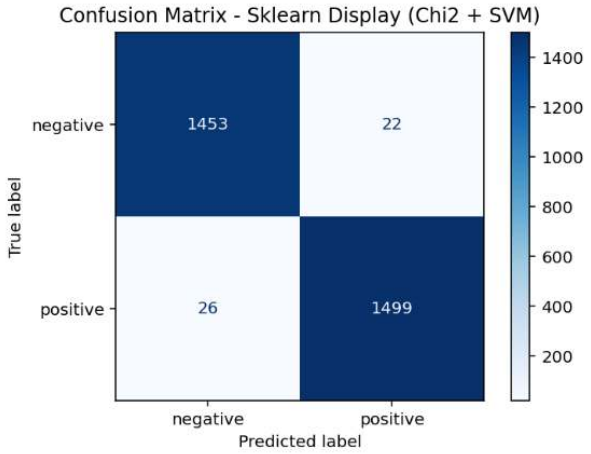


Şekil 16. PCA + Random Forest Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

Bu model, metin sınıflandırma görevinde TF-IDF, Ki-kare (Chi2) özellik seçimi ve SVM algoritmalarını birleştirerek test verileri üzerinde %98.40'lık oldukça yüksek bir doğruluk oranı elde etmiştir. Bu sonuç, öncelikle TF-IDF ile metinlerin kelime ve ikili kelime grupları (n-gram aralığı 1-2) düzeyinde sayısal özelliklere dönüştürülmesinin, ardından Ki-kare testi ile sınıflandırma görevi için en ayırt edici 1000 özelliğin seçilmesinin ve son olarak SVM algoritmasının bu seçilmiş özellikler üzerinde etkili bir şekilde sınıflandırma yapmasının birleşimiyle ortaya çıkmıştır.

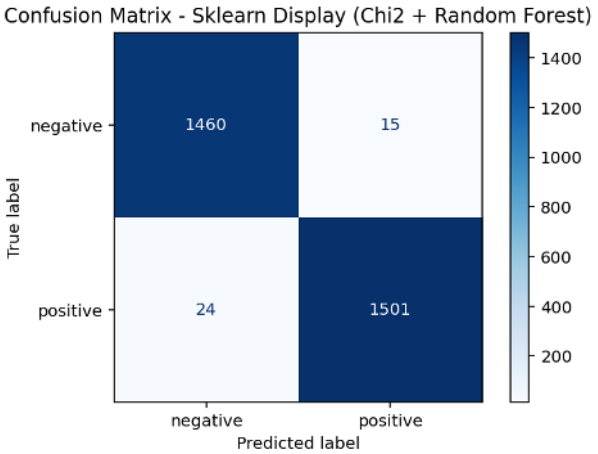
Ki-kare özellik seçimi, gürültülü veya sınıflandırma için az bilgi içeren özellikleri filtreleyerek modelin daha odaklanmış ve yüksek performanslı çalışmasına olanak tanımış olabilir. Elde edilen yüksek doğruluk oranı ve Ki-kare analizine göre belirlenen en anlamlı kelimeler, bu yöntemin metin sınıflandırma problemleri için güçlü ve yorumlanabilir bir yaklaşım sunduğunu göstermektedir.



Şekil 17. Ki-Kare + SVM Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

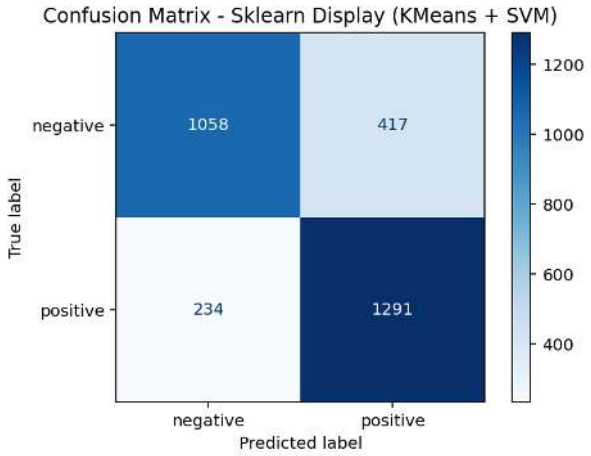
Bu deneyde, metin verilerini sınıflandırmak amacıyla TF-IDF, Ki-kare (Chi2) özellik seçimi ve Random Forest algoritmaları ardışık olarak kullanılmıştır. Elde edilen sonuçlara göre, bu birleşik model test verileri üzerinde %98.70'lik oldukça yüksek bir doğruluk oranına ulaşmıştır. Bu başarı, öncelikle TF-IDF yöntemiyle metinlerin kelime ve ikili kelime grupları (n-gram aralığı 1-2) düzeyinde sayısal özelliklere dönüştürülmesinin, ardından Ki-kare testi ile sınıflandırma görevi için en ayırt edici 1000 özelliğin seçilmesinin ve son olarak Random Forest algoritmasının bu seçilmiş özellikler üzerinde etkili bir sınıflandırma performansı sergilemesinin bir sonucudur. Ki-kare özellik seçimi, modelin performansını artırmak için en relevant özellikleri belirlemede önemli bir rol oynamış olabilir. Ayrıca, elde edilen en anlamlı 20 kelime ve bunların Ki-kare skorları, modelin hangi kelimeleri sınıflandırma kararlarında en etkili bulduğu hakkında değerli bilgiler sunmaktadır. Bu yüksek doğruluk oranı ve anlamlı özelliklerin belirlenmesi, bu yöntemin metin sınıflandırma problemleri için güçlü ve yorumlanabilir bir yaklaşım olduğunu desteklemektedir.



Şekil 18. Ki-Kare + Random Forest Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

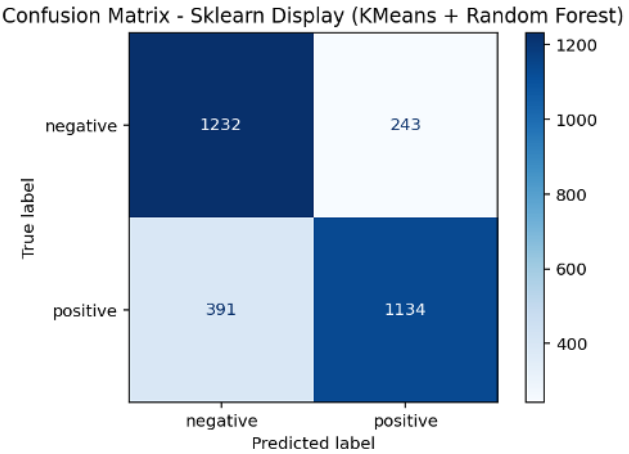
Bu deneyde, metin sınıflandırması için TF-IDF, K-Means kümeleme ve Destek Vektör Makineleri (SVM) algoritmaları bir arada kullanılmıştır. Elde edilen sonuçlara göre, bu birleşik model test verileri üzerinde %78.30'luk bir doğruluk oranı sergilemiştir. Bu yaklaşımda, öncelikle metin verileri TF-IDF ile kelime ve ikili kelime grupları (n-gram aralığı 1-2) düzeyinde sayısal vektörlere dönüştürülmüş, ardından bu vektörler K-Means algoritması ile 5 farklı kümeye ayrılmıştır. K-Means'ten elde edilen uzaklık bilgisi, SVM sınıflandırıcısı için ek bir özellik uzayı olarak kullanılmıştır. Son olarak, SVM algoritması bu özellikler üzerinde sınıflandırma işlemini gerçekleştirmiştir. Elde edilen %78.30'luk doğruluk oranı, K-Means ile oluşturulan kümelerin metin sınıflandırması için doğrudan yüksek bir ayırt edicilik sağlamadığını göstermektedir. Ancak, K-Means'in metin verisindeki potansiyel anlamsal gruplamaları yakalama çabası, SVM'in karar sınırlarını oluşturmasında dolaylı bir etkiye sahip olabilir. Bu deneysel sonuç, K-Means tabanlı özellik mühendisliğinin metin sınıflandırma problemlerinde dikkatli bir şekilde değerlendirilmesi gerektiğini işaret etmektedir.



Şekil 19. Kmeans + SVM Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

Bu deneyde, metin sınıflandırması için TF-IDF, K-Means kümeleme ve Random Forest algoritmaları bir arada kullanılmıştır. Elde edilen sonuçlara göre, bu birleşik model test verileri üzerinde %78.87'lik bir doğruluk oranı göstermiştir. Bu yaklaşımda, öncelikle metin verileri TF-IDF ile kelime düzeyinde sayısal vektörlere dönüştürülmüş, ardından bu vektörler K-Means algoritması ile 5 farklı kümeye ayrılmıştır. K-Means'ten elde edilen uzaklık bilgisi, Random Forest sınıflandırıcısı için ek bir özellik uzayı olarak kullanılmıştır. Son olarak, Random Forest algoritması bu özellikler üzerinde sınıflandırma işlemini gerçekleştirmiştir. Elde edilen %78.87'lik doğruluk oranı, K-Means tarafından oluşturulan kümelerin tek başına güçlü bir sınıflandırma sinyali sunmadığını düşündürmektedir. Ancak, Random Forest'ın karmaşık karar sınırları oluşturabilme yeteneği, K-Means tarafından yakalanan potansiyel anlamsal yapıları bir miktar değerlendirmiş olabilir. Bu sonuçlar, K-Means tabanlı özellik mühendisliğinin, özellikle Random Forest gibi esnek modellere entegre edildiğinde, metin sınıflandırma problemlerinde dikkatli bir şekilde incelenmesi gerektiğini göstermektedir.



Şekil 20. Kmeans + Random Forest Modelinin Confusion Matrix Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

Bu çalışmada, metin sınıflandırma görevi için farklı makine öğrenimi algoritmaları ve özellik mühendisliği tekniklerinin performansları detaylı bir şekilde incelenmiştir. Elde edilen test sonuçları, kullanılan yöntemlerin doğruluk oranları ve çalışma süreleri açısından önemli farklılıklar gösterdiğini ortaya koymuştur.

Tek başına uygulanan TF-IDF ve Destek Vektör Makineleri (SVM) modeli, %97.70'lik bir doğruluk oranıyla dikkat çekmiş ve 7.52 saniyelik bir çalışma süresiyle kabul edilebilir bir verimlilik sunmuştur. Bu sonuç, SVM'in metin verisinin karmaşık yapısını modellemedeki etkinliğini bir kez daha teyit etmektedir.

Benzer şekilde, sadece sınıflandırıcı olarak TF-IDF ve Random Forest kombinasyonu da %97.10'luk yüksek bir doğruluk oranına ulaşmış ve 7.42 saniyelik çalışma süresiyle yine süre olarak kabul edilebilir bir verimlilik sunmuştur. Bu, Random Forest'ın metin sınıflandırma problemlerinde SVM kadar olmasa da yüksek doğruluk yeteneği sunduğunu göstermektedir.

TF-IDF ve Naive Bayes modeli ise %93.73'lük bir doğruluk oranıyla diğerlerine göre daha düşük bir performans sergilemiş, ancak 1.45 saniyelik en kısa çalışma süresine sahip olmuştur. Bu durum, Naive Bayes'in basitliği ve hızıyla öne çıkmasına rağmen, metin verisindeki karmaşık ilişkileri yakalamada sınırlı kalabileceğini işaret etmektedir.

Boyut indirgeme tekniklerinin entegrasyonu da önemli sonuçlar doğurmuştur. TF-IDF, Temel Bileşenler Analizi (PCA) ve SVM kombinasyonu %98.30'luk en yüksek doğruluk oranlarından birine ulaşmış, ancak 7.04 saniyelik çalışma süresiyle nispeten daha fazla işlem gücü gerektirmiştir. TF-IDF, PCA ve Random Forest modeli ise %95.30'luk bir doğrulukla daha düşük bir performans göstermiş ve 5.39 saniyelik çalışma süresiyle sadece random forest kullanılan modele kıyasla daha düşük bir çalışma süresine sahip olmuştur. PCA'nın Random Forest ile birlikte kullanımının, işlem süresinde azalmaya sebep olsa da bu özel veri setinde performansı artırmadığı görülmektedir.

Özellik seçimi yöntemlerinin kullanımı da kayda değer sonuçlar sunmuştur. TF-IDF, Ki-kare (Chi2) özellik seçimi ve SVM modeli %98.40'lık yüksek bir doğruluk oranıyla öne çıkarken, 2.36 saniyelik çalışma süresiyle de verimli bir denge sunmuştur. Benzer şekilde, TF-IDF, Ki-kare özellik seçimi ve Random Forest kombinasyonu %98.70'lik en yüksek doğruluk oranına ulaşmış ve 2.28 saniyelik makul bir çalışma süresine sahip olmuştur. Bu sonuçlar, Ki-kare özellik seçiminin hem SVM hem de Random Forest modellerinin performansını artırmada etkili bir yöntem olduğunu göstermektedir.

Kümeleme tabanlı özellik mühendisliği yaklaşımları ise daha farklı sonuçlar ortaya koymuştur. TF-IDF, K-Means kümeleme ve SVM modeli %78.30'luk bir doğruluk oranıyla en düşük performanslardan birini sergilemiş ve 1.60 saniye çalışmıştır. Benzer şekilde, TF-IDF, K-Means kümeleme ve Random Forest modeli de %78.87'lik bir doğruluk oranıyla düşük bir performans göstermiş, ancak 1.04 saniyelik en kısa çalışma sürelerinden birine sahip olmuştur. Bu sonuçlar, K-Means algoritmasının bu özel metin sınıflandırma görevi için doğrudan etkili özellikler üretmediğini düşündürmektedir.

Elde edilen sonuçlar ışığında, gelecekteki çalışmalar ve uygulamalar için aşağıdaki öneriler sunulmaktadır:

Yüksek Doğruluk Odaklı Uygulamalar İçin: Metin sınıflandırma doğruluğunun kritik olduğu uygulamalarda, TF-IDF ile birlikte Ki-kare özellik seçimi ve ardından SVM veya Random Forest algoritmalarının kullanılması en uygun yaklaşımlar olarak öne çıkmaktadır. Özellikle TF-IDF + Chi2 + Random Forest kombinasyonu, en yüksek doğruluk oranını (%98.70) sunmaktadır.

Hızlı ve Verimli Uygulamalar İçin: İşlem hızının önemli olduğu senaryolarda, TF-IDF, Ki-Kare testi ve Random Forest kombinasyonu (%98.70 doğruluk, 2.28 saniye) veya TF-IDF ve Ki-kare özellik seçimi ile birlikte SVM (%98.40 doğruluk, 2.36 saniye) iyi bir denge sunmaktadır. Naive Bayes ise en hızlı seçenek olmasına rağmen, doğruluk performansı diğerlerine göre daha düşüktür.

Boyut İndirgeme Tekniklerinin Dikkatli Kullanımı: PCA gibi boyut indirgeme tekniklerinin metin sınıflandırma performansına etkisi veri setine ve kullanılan algoritmaya bağlı olarak değişebilmektedir. Bu çalışmada, PCA'nın işlem süresini kısaltsa da bazı durumlarda doğruluğu artırmadığı gözlemlenmiştir. Bu nedenle, boyut indirgeme teknikleri uygulanmadan önce dikkatli bir analiz yapılması ve farklı parametrelerle denenmesi önerilmektedir.

Kümeleme Tabanlı Yaklaşımların Alternatif Kullanımları: K-Means kümelemesinin doğrudan sınıflandırma doğruluğunu artırmadığı görülmüştür. Ancak, kümeleme algoritmaları, etiketlenmemiş veriyi keşfetmek, veri ön işleme adımlarında veya farklı özellik mühendisliği yaklaşımlarında potansiyel olarak faydalı olabilir. Gelecekteki çalışmalar, kümeleme sonuçlarının sınıflandırma performansını dolaylı yollarla nasıl iyileştirebileceğine odaklanabilir.

Daha Gelişmiş Modellerin Araştırılması: Derin öğrenme tabanlı modeller (örneğin, Transformer ağları) gibi daha karmaşık algoritmalar, metin verisindeki bağlamsal bilgiyi daha iyi yakalayarak daha yüksek doğruluk oranları sunma potansiyeline sahiptir. Gelecek çalışmalar, bu tür gelişmiş modellerin bu veri seti üzerindeki performansını araştırmalıdır.

Özellik Mühendisliğinin Derinleştirilmesi: Temel TF-IDF vektörleştirmesinin ötesine geçilerek, kelime gömlekleri (word embeddings), belge gömlekleri (document embeddings) veya n-gram aralığının optimizasyonu gibi daha gelişmiş özellik mühendisliği teknikleri denenerek model performansının artırılması hedeflenebilir.

Bu çalışma, farklı makine öğrenmesi algoritmaları ve özellik mühendisliği yaklaşımlarını karşılaştırmalı olarak ele alarak, metin sınıflandırma problemlerinde hangi yöntemlerin daha etkili sonuçlar verdiğine dair kapsamlı bir değerlendirme sunmaktadır. Özellikle TF-IDF ile birlikte Ki-kare özellik seçimi ve Random Forest algoritmasının bir arada kullanılmasıyla elde edilen %98.70'lik doğruluk oranı, literatürde yüksek doğruluk arayışındaki çalışmalara önemli bir katkı sağlamaktadır. Ayrıca kodun çalışma süresi ve doğruluk arasında denge arayan uygulamalar için de öneriler geliştirilmiş, bu yönüyle hem akademik hem de pratik uygulamalar açısından yol gösterici bir çalışma olmuştur.

Bu çalışmanın bazı sınırlılıkları bulunmaktadır. Öncelikle, yalnızca Kaggle platformundan elde edilen belirli bir İngilizce Twitter veri seti kullanılmıştır ve bu durum, sonuçların genelleştirilebilirliğini sınırlayabilir. Ayrıca, çalışma kapsamında yalnızca belirli makine öğrenmesi algoritmaları ve temel özellik mühendisliği teknikleri değerlendirilmiştir; derin öğrenme tabanlı daha gelişmiş modeller bu araştırma kapsamında yer almamıştır.

Gelecek çalışmalar için, daha büyük ve farklı sosyal medya platformlarından elde edilecek veri setleri üzerinde benzer analizlerin gerçekleştirilmesi önerilmektedir. Ayrıca derin öğrenme tabanlı modellerin (örneğin, BERT, RoBERTa, Transformer tabanlı modeller) bu veri setleri üzerinde performansının değerlendirilmesi, metin sınıflandırma doğruluğunun artırılmasına katkı sağlayabilir. Son olarak, daha ileri özellik mühendisliği yöntemleri, kelime gömlekleri veya cümle temelli temsil öğrenme yaklaşımlarının da incelenmesi, bu alanda yeni katkılar sunabilir.

Sonuç olarak, bu çalışma farklı makine öğrenimi yaklaşımlarının metin sınıflandırma görevindeki etkinliğini kapsamlı bir şekilde değerlendirmiştir. Elde edilen sonuçlar, özellik seçimi yöntemlerinin (özellikle Ki-kare) ve güçlü sınıflandırıcıların (SVM ve Random Forest) bu tür problemler için yüksek performans sağladığını göstermektedir. Gelecekteki araştırmalar, daha gelişmiş modellerin ve özellik mühendisliği tekniklerinin potansiyelini keşfederek metin sınıflandırma doğruluğunu daha da artırmaya odaklanmalıdır.

KAYNAKÇA

İlhan, N., ve Sağaltıcı, D. (2020). Twitter'da duygu analizi. *Harran Üniversitesi Mühendislik Dergisi*, *5*(2), 146–156. https://doi.org/10.46578/humder.772929

Demir, Ö., Baban Chawai, A. I., ve Doğan, B. (2019). Türkçe metinlerde sözlük tabanlı yaklaşımla duygu analizi. *International Periodical of Recent Technologies in Applied Engineering*, *2*, 58–66. https://doi.org/10.35333/porta.2019.98

Kandıran, E., Gümüş, B., ve Özer, M. A. (2022). Covid-19 pandemi sürecinde uzaktan eğitimin Twitter yansımalarının duygu analizi. *International Journal of Social Sciences and Education Research*, *8*(3), 228–242. https://doi.org/10.24289/ijsser.1102248

Kılıç, G., Budak, İ., ve Kılıç, B. S. (2020). Kara Cuma etiketlerinin tweet istatistikleri ve duygu analizi ile sıralanması. *Selçuk Üniversitesi Sosyal Bilimler Meslek Yüksekokulu Dergisi*, *23*(1), 131–140. https://doi.org/10.29249/selcuksbmyd.638064

Tuzcu, S. (2020). Çevrimiçi kullanıcı yorumlarının duygu analizi ile sınıflandırılması. *ESTUDAM Bilişim Dergisi*, *1*(2), 1–5.

Akın, B., ve Gürsoy Şimşek, U. T. (2018). Sosyal medya analitiği ile değer yaratma: Duygu analizi ile geleceğe yönelim. *Mehmet Akif Ersoy Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, *5*(3), 797–811. https://doi.org/10.30798/makuiibf.435804

Erkırtay, O. Ş., ve Ünal, C. (2021). Toplum çevirmenliğinde fikir madenciliği ve duygu analizi. *Manisa Celal Bayar Üniversitesi Sosyal Bilimler Dergisi*, *19*(3), 168–185. https://doi.org/10.18026/cbayarsos.890384

Kızılkaya, Y. M. (2018). *Duygu analizi ve sosyal medya alanında uygulama* [Doktora tezi]. Uludağ Üniversitesi Sosyal Bilimler Enstitüsü.

TOPLUMSAL KATKI

Bu araştırma, dijital çağın hızla büyüyen sosyal medya ortamlarında kullanıcıların duygu durumlarını tespit etmenin zorluklarını ele almakta ve bu doğrultuda, makine öğrenmesi tabanlı etkili duygu analiz yöntemleri geliştirmeyi amaçlamaktadır. Sosyal medya platformlarında üretilen büyük hacimli veri, bireylerin düşüncelerini, kaygılarını ve taleplerini içeren önemli bir kaynak olmasına rağmen, bu verinin anlamlandırılması ve analiz edilmesi noktasında hâlen ciddi yapısal ve teknik eksiklikler bulunmaktadır. Özellikle bireylerin toplumsal olaylara, politik gelişmelere veya kriz dönemlerine ilişkin tepkilerinin hızlı ve doğru şekilde analiz edilememesi; gerek kamu politikalarının şekillendirilmesinde gerekse kriz yönetiminde gecikmelere yol açabilmektedir.

Bu tez çalışması, söz konusu probleme çözüm üretmek amacıyla doğal dil işleme ve makine öğrenmesi tekniklerini birleştirerek sosyal medya metinlerinde duygu analizi gerçekleştiren, doğruluk oranı yüksek sınıflandırma ve tahminleme modelleri sunmaktadır. Geliştirilen modeller, özellikle kamu kurumları, medya kuruluşları ve sosyal araştırmacılar için sosyal medya eğilimlerini analiz etme noktasında güçlü bir araç sunmaktadır. Bu sayede, toplumun belirli konulara yönelik pozitif veya negatif duygu durumları ölçülebilir, kamuoyunun tepkileri daha sağlıklı biçimde değerlendirilebilir ve politika yapıcılar için veri temelli karar alma süreçleri desteklenebilir.

Araştırmada hibrit modellerin geleneksel sınıflandırıcılara göre daha yüksek doğruluk oranları sunduğu ortaya konmuş ve bu da, büyük veri analizinde hem doğruluk hem de verimliliği artıran yeni yaklaşımların mümkün olduğunu göstermiştir. Uygulamada kullanılan TF-IDF, Chi-Square, PCA, KMeans gibi tekniklerin sosyal medya verisine özel uyarlanması sayesinde, farklı bağlamlardaki duygu kalıpları daha iyi ayrıştırılmış ve sosyal medya içeriklerinin daha doğru sınıflandırılması sağlanmıştır.

Bu çalışma, aynı zamanda akademik alanda duygu analizi yöntemleriyle ilgilenen araştırmacılar için kapsamlı bir teknik referans sunarken, yazılım geliştiriciler ve veri bilimi uzmanlarına da yeniden kullanılabilir ve geliştirilebilir bir model altyapısı kazandırmaktadır. Modelin açık kaynak kütüphanelerle oluşturulmuş olması, erişilebilirliğini artırmakta ve gelecekte farklı alanlara (örneğin müşteri yorum analizi, kamuoyu yoklamaları, kriz izleme sistemleri) uygulanabilirliğini mümkün kılmaktadır.

Uzun vadede, bu araştırmanın toplumsal katkısı; sosyal medya gibi yüksek etkili iletişim ortamlarında ifade edilen duygu ve düşüncelerin daha doğru analiz edilmesiyle toplumun nabzının daha gerçekçi biçimde tutulmasına olanak sağlamaktır. Böylece, toplumsal gelişmelerin daha doğru anlaşılması, hızlı tepki verilmesi gereken olaylarda kamu kurumlarının zamanında aksiyon alması, halkla ilişkilerin daha duyarlı biçimde yürütülmesi gibi pek çok alanda fayda sağlanacaktır.

Sonuç olarak, bu çalışma yalnızca teknik bir model üretmenin ötesine geçerek, duygu analizi yoluyla toplumun düşünsel reflekslerinin anlaşılması, toplumsal bilinç düzeyinin takip edilmesi ve kamu politikalarının şekillendirilmesine katkı sunabilecek bir altyapı sağlamaktadır. Bu yönüyle tez, hem akademik hem de toplumsal düzeyde anlamlı ve sürdürülebilir bir katkı sunma potansiyeline sahiptir.