İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
FEN BİLİMLERİ ENSTİTÜSÜ

Ürün Değerlendirmeleri Üzerinde Duygu Analizi için Makine Öğrenmesi ve Derin Öğrenme Metotlarının Karşılaştırılması

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Mehmet Emrah DEMİRAĞ

ORCID 0009-0003-0765-8470

Proje Danışmanı: Prof. Dr. Aytuğ ONAN

Şubat 2024

Ürün Değerlendirmeleri Üzerinde Duygu Analizi için Makine Öğrenmesi ve Derin Öğrenme Metotlarının Karşılaştırılması

Öz

İnternetin kullanımı; buna bağlı olarak da e-ticaretin önemi gün geçtikçe artmaktadır. E-ticaret sistemini kullanan kişiler, ilgili e-ticaret sitelerinde satın aldıkları ürünler hakkında yorumlar yapabilmektedirler. Bu yorumlar, pek çok başka mecrada yapılan yorumlarda olduğu gibi, internet üzerinde oluşan bir veri yığınına katkıda bulunmaktadır. Bu yığının artması, araştırmacılar için zengin bir kaynak oluştururken; aynı zamanda e-ticaretin ilerlediği yolda bir gösterge olabilmesi için analiz edilmesi gereken bir bilgi birikimine işaret etmektedir.

Bu çalışmada Amerika Birleşik Devletleri merkezli Amazon.com sitesinde yer alan çeşitli elektronik aletler için yapılan kullanıcı yorumları, web kazıma yöntemi ile 50.000 satırlık bir veri kümesi halinde elde edilmiş; bu veri kümesi, çeşitli ön işlemlerden geçerek, duygu analizi kapsamında, yorumların olumlu veya olumsuz oluşunun tespiti için 5 farklı makine öğrenmesi ve 4 farklı derin öğrenme algoritmasına tabi tutulmuştur. Çeşitli karşılaştırma süreçleri sonucunda birincil süreçte %85,85, ikincil süreçte ise %90 ile en yüksek doğruluk oranına ulaşan Geçitli Tekrarlayan Birimler (GRU) algoritması; duygu analizinde derin öğrenme metotlarının, makine öğrenmesi algoritmalarına göre daha başarılı olduğu sonucuna varılmasını sağlamıştır.

Anahtar Sözcükler: Duygu Analizi, Metin Madenciliği, Makine Öğrenmesi, Derin Öğrenme, Web Kazıma, Ürün Değerlendirmeleri, E-Ticaret.

Comparison of Machine Learning and Deep Learning Methods for Sentiment Analysis on Product Reviews

Abstract

The use of the Internet and therefore the importance of e-commerce is increasing day by day. People who use the e-commerce system can comment on the products they purchased on relevant e-commerce sites. These comments, like comments made in many other media, contribute to a mass of data on the internet. While increasing this pile creates a rich resource for researchers; it also points to a body of knowledge that needs to be analyzed to be an indicator of the path that e-commerce is progressing.

In this study, user comments for various electronic devices on the United States-based Amazon.com site were obtained by the web scraping method. After a dataset of 50,000 lines had been obtained; this data set, after going through various pre-processing for sentiment analysis purposes, was tested by 5 different machine learning and 4 different deep learning algorithms to determine whether the comments were positive or negative. As a result of various comparison processes, the Gated Recurrent Units (GRU) algorithm reached the highest accuracy rate of 85.85% in the primary process and 90% in the secondary process. This study has led to the conclusion that deep learning methods are more successful than machine learning algorithms in the sentimental analysis area.

Keywords: Sentimental Analysis, Text Mining, Machine Learning, Deep Learning, Web Scraping, Product Reviews, E-Commerce.

İçindekiler

Şekiller Listesi

Tablolar Listesi

Bölüm 1

Giriş

Dijitalleşme kavramının duyulduğunda yabancılık çekilmediği, insanlığın yaşam tarzını sürekli değiştiren; insanların her geçen gün makinelerin daha fazla yardımını aldığı bir dünya düzeni hâkim sürmektedir. Bu dünya düzeni, "yeni ekonomi" adı verilen; fiziksel faktörlerden çok bilgisayar sistemlerine dayalı bir ekonomi modelinin her geçen gün daha fazla hayatımıza nüfuz ettiği gerçeğiyle perçinlenmektedir (Afşar, 2001).

Ticaret, çok geniş bir bakış açısıyla; ekonomik değer taşıyan mal veya ürünlerin üretildiği andan tüketildiği ana kadar yaşadığı değiş tokuş sürecidir. E-ticaret ise tüm bu işlemlerin internete bağlı platformlarda gerçekleşmesidir (Bayrak, 2023). Günümüzde kabaca "internetten alışveriş yapmak" olarak tanımlanan e-ticaret, satın alma alışkanlıklarını tamamen değiştirmektedir. Bir şehirde ya da daha küçük bir nüfus biriminde, bir köyde yaşayan, belirli bir sosyal çevresi olan bir kişi; akrabalarından başlayıp arkadaşlarına sorular sorarak bir anket düzenlese; çok büyük bir olasılıkla, tanıdıkları arasından internetten alışveriş yapan en az bir kişi tespit edebilecektir. Ülkemizde Hepsiburada, Trendyol, YemekSepeti gibi çok farklı uzmanlıklarda faaliyet gösteren e-ticaret siteleri mevcuttur. Fakat e-ticaretin ilk temsilcilerinden olan ve 1994 yılında kurulmuş olan Amazon.com sitesi, ilk zamanlarında kitap satışı ile faaliyetlerine başlamış olsa da bugün, çok geniş bir yelpazede ürün satışı gerçekleştiren ve dünyanın en büyük şirketlerinden biri haline gelmiştir[1].

E-ticaret yapılan platformlarda; kişilerin ilgili mal ve/veya hizmetler, yerlerin özellikleri gibi bilgileri başkalarıyla paylaşabildikleri, kendilerini ifade edebildikleri, "yorumlama ve puanlamaya dayalı" mekanizmalar bulunmaktadır. Kullanıcılar, gittikleri restoranlardaki yemeklerin ne kadar "lezzetli/harika" ya da ne kadar

1 Wikipedia.org; "AMAZON (COMPANY); "https://en.wikipedia.org/wiki/Amazon_(company); Erişim: 09.06.2023

“kötü/berbat” olabildiği üzerine özgürce yorum yapabilmekte; ayrıca bunları puanlama yaparak destekleyebilmektedirler. Kişiler, konakladıkları otellerle ilgili bilgiler sayesinde başka kişileri gerek doğru gerekse yanlış bir şekilde yönlendirebilmektedir. İnsanların da bu yapılan yorumlara büyük bir çoğunlukla güvendikleri; tatil planlarını, restoran tercihlerini, ziyaret aşamalarını bu yorum ve puanlamalara göre yapabildikleri, bilinen bir durumdur. Kişiler; Amazon gibi, hepsiburada.com gibi, eBay gibi sitelerdeki puan ve yorumlamalara göre ürünleri inceleyebilmekte ve birbirine rakip markaların bu sitelerde beraber yer aldığı ürünlerini, tamamen bu puanlama ve incelemelere dayalı seçip satın alabilmektedirler.

Günümüzde Facebook, Twitter, Instagram, TikTok, LinkedIn gibi dünyada internet olan neredeyse her noktaya ulaştığını tahmin edebileceğimiz platformlar faaliyet göstermektedir. Sosyal medya üzerinden siyasi seçimler öncesinde pek çok yorumun yapıldığı; kitlelerin etkilenmeye çalışıldığı, bu platformların kullanıcıları tarafından rahatlıkla gözlemlenebilecek bir durumdur. Pek çok ülkedeki politikacıların, bürokratların, siyasi partilerin ve hatta kurumların sosyal medya hesaplarının olduğu; bu kişi ve kurumların, hitap ettikleri kitlelere bildirim, bilgilendirme, hatta propaganda iletimlerinde bulunduğu gözlemlenebilir.

Gerek sosyal medya gerekse e-ticaret olsun; akademik veya özel sektör mensubu araştırmacıların, internet üzerindeki bu trafik sonucunda elde edebilecekleri bir veri yığını söz konusudur. Bu veriler dinamiktir; çünkü her an oluşturulmaya devam edilmektedir. Sürekli büyüyen bu veri yığının içinden, ilgilenilmek istenen araştırma konularıyla ilgili olanların elde edilip, derlenip çeşitli araştırmalar yapılması; bu araştırmalar sonucunda çeşitli çıkarımlara varılması mümkündür.

Bu çalışma, Amerika Birleşik Devletleri merkezli Amazon.com sitesindeki çeşitli elektronik ürünler üzerine yapılan kullanıcı yorumlarının incelenerek; bu yorumlarda kullanılan kelimelerin, kullanıcıların verdikleri puanlamalara göre kendilerini ifade ettikleri duygularla ne kadar örtüştüğünün tespit edilme isteği üzerine ortaya çıkmıştır. Çalışmanın gerçekleştirilmesi için “duygu analizi” adı verilen bir yöntem kullanılmıştır. Bu analiz gerçekleştirilirken “makine öğrenmesi” ve “derin öğrenme” adı verilen metotlardan ve bu metotlara bağlı çeşitli algoritmalardan yararlanılması gerekmiştir. Sitedeki çeşitli ürünlere dayalı yapılmış, toplamda 50.000 satırdan oluşan bir veri kümesi; yorum için 5 üzerinden yapılan puanlama ve yorumun içeriği

açısından ele alınarak, üzerinde beş makine öğrenmesi yöntemi (K – En Yakın Komşu, Lojistik Regresyon, Naive Bayes, Rastgele Ormanlar ve Destek Vektör Makineleri) ve dört derin öğrenme algoritması (Geçitli Tekrarlayan Birimler, Tekrarlayan Sinir Ağı, Geçitli Tekrarlayan Birimler ve Uzun Kısa-Süreli Hafıza) uygulanarak analiz edilmeye çalışılmıştır. Yapılan bu işlemler sonucunda, derin öğrenme algoritmaları arasından "Geçitli Tekrarlayan Birimler" algoritmasının en yüksek doğruluk değerlerine ulaştığı; kullanıcıların "olumlu" ya da "olumsuz" duygularla yazdıkları yorumları sınıflandırmada başarı potansiyeli en yüksek algoritma olduğu sonucuna varılmıştır.

Bu çalışmanın takip eden bölümlerinde şu konular irdelenmektedir: İkinci bölümde duygu analizi ile ilgili yapılmış başka çalışmalar kısaca incelenmiştir. Üçüncü bölümde bu çalışmanın metodolojisinden; veri kümesinin nasıl elde edildiğinden, web kazıma kavramı ve nasıl gerçekleştirildiğinden; verinin ön işlemesi ve analize hazır hale getirilişinden; son olarak da analizin nasıl yapıldığı ve analiz sonucunda elde edilen verilerden bahsedilerek ilgili süreç incelenmiştir. Dördüncü Bölümde ise söz konusu süreç sonucu erişilen sonuçlar ile ilgili yorumlar yer almaktadır.

Bölüm 2

İlgili Çalışmalar

Bir önceki bölümde de belirtildiği üzere; internet üzerinde her an oluşmaya devam eden veri yığını, araştırmacılar için zengin bir kaynak oluşturmaya devam etmektedir. Bundan dolayı, farklı disiplinlerdeki gerek özel sektör mensubu gerekse akademik araştırmacılar tarafından gerçekleştirilmiş pek çok çalışma bulunmaktadır. Bu çalışma kapsamında faaliyetlerden bahsetmeden önce, ilgili araştırmacıların gerçekleştirmiş olduğu çalışmalar arasından alınan, farklı bakış açıları içeren beş örnekten bahsetmekte yarar vardır.

2.1. Türkçe Metinler İçin Duygu Analizi Yaklaşımı İle İletişimde Bağlamdan Bağımsız Modellerin Geliştirilmesi Üzerine Bir Araştırma: Karma Veri Modeli Önerisi

Marmara Üniversitesi'nden Çiğdem AYTEKİN ve Mehmet Ali BAYRAM, "Yeni Medya Elektronik Dergisi" adlı Dergide yayımlanan "Türkçe Metinler İçin Duygu Analizi Yaklaşımı İle İletişimde Bağlamdan Bağımsız Modellerin Geliştirilmesi Üzerine Bir Araştırma: Karma Veri Modeli Önerisi" isimli araştırma makalelerinde; internet kullanıcılarının satın aldıkları/tükettikleri ürün ve hizmetler konusunda ilgili mecralara yazdıkları olumlu veya olumsuz yorumlara odaklanmışlardır. Bu yorumların, ilgili platformun sahibi olan e-ticaret şirketi açısından yarattığı itibar ve bunun yönetimi vb. durumlar açısından değerlendirilmesi gerektiğine; bununla beraber ilgili yorumların, diğer kullanıcıları da etkilediğine, zira bu diğer kullanıcıların bir alım esnasında ilgili yorumlara ulaşarak görüşleri değerlendirdiklerine ve onları referans olarak kullandıklarına dikkat çekmişlerdir. İlgili yorumların hacminin büyümesi nedeniyle otomatik olarak analiz edilmeleri gerektiğini ve duygu analizinin bu amaçla sıklıkla kullanılan bir yöntem olduğunu ifade etmişlerdir. Araştırmalarının amacını,

“Duygu Analizi ve makine öğrenmesi yöntemleri ile iletişimde bağlamın etkisini ortaya koymak ve bu etkiyi ortadan kaldıracak bağlamdan bağımsız modellerin geliştirilmesi için bir Karma Veri Uygulaması önerisinde bulunmak” olarak açıklamışlardır (Aytekin ve Bayram, 2021).

AYTEKİN ve BAYRAM; yapılan yorumların, yapıldıkları mecralara dayalı olarak analiz edilmesi gerektiği görüşü üzerine eğilmiş; hem ürün, film ve kitap yorumlamalarının kendilerine ait farklı duygu analizi modelleriyle analiz edildiğinde daha başarılı sonuçlar aldığına hem de her mecraya ait model kullanımının pratik olmayışına dikkat çekmiş; bu nedenle de ortak bir model oluşturularak ortak analiz gerçekleştirilebilmesi fikri üzerine yoğunlaşmışlardır. Çalışmalarında hem Twitter hem de diğer mecralar (film, haber, kitap siteleri vs.) bazlı yapılmış başka duygu analizi üzerine çalışmalardan da bahsetmişlerdir.

AYTEKİN ve BAYRAM; yorumbudur.com, hepsiburada.com ve beyazperde.com sitelerindeki kullanıcı yorumlarını urllib3 adlı Python kütüphanesini kullanarak indirmiş, bunları bir csv dosyasında birleştirmişlerdir. Jetonlaştırma (Tokenization) adlı yöntemi kullanarak en çok tekrar eden belli sayıda kelimeyi seçerek bir liste oluşturmuşlar; bu listeyi kullanarak Python Sklearn kütüphanesi ile rastgele karıştırdıkları veri kümesini 90:10 oranlarında eğitim ve test kümesi olarak bölerek sıralı derin öğrenme mimarisiyle eğitip test etmişlerdir. Site türlerine göre kendi mecraları ve diğer mecralarla karşılaştırılan veriler; kendi modelleriyle karşılaştırıldıktan sonra Karma Veri Uygulaması modeli ile karşılaştırılmış; yorumların olumluluğu/olumsuzluğunun tespiti konusunda “doğruluk oranları” (accuracy) incelendiğinde önerilen karma modelin başarılı olduğu sonucuna varmışlardır (Aytekin ve Bayram, 2021).

2.2. Metin Madenciliği Ve Duygu Analizi İle Siber Zorbalık Tespiti

Eskişehir Osmangazi Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü’nden Elif Şevval DİNÇER, Duygu KAYAOĞLU ve Simara SAFARLI, “Metin Madenciliği ve Duygu Analizi ile Siber Zorbalık Tespiti” adlı çalışmalarında; sosyal medya iletişiminin olumsuz yanlarından biri olan, "gerçek

hayatta insanların başkalarına söylediklerinde karşı taraftan büyük tepki alabilecekleri ifadeleri, kimliklerini gizli tutarak karşı tarafa, onları incitmeyi ve kırmayı amaçlayarak yöneltme" veya "bilgi teknolojilerinden faydalanan kişi veya kişilere veya tüzel kişiliklere yönelik teknik veya ahlaki, zarar verme amacını taşıyan eylemlerin geneli" veya "elektronik metin ya da metinlerle sınırlı ve tekrar eden zarar/hasar" olarak ayrı ayrı tanımladıkları "Siber Zorbalık" kavramına odaklanmışlardır. Siber zorbalığın, sosyal açıdan, özellikle genç insanları, ergenleri nasıl olumsuz etkilediğine dikkat çekmişlerdir.

Twitter Application Programming Interface (API) kullanarak Twitter platformu üzerinden veriler elde etmişler ve bu verileri düzenleyerek Destek Vektör Makinesi (SVM), Lojistik Regresyon (LR), Naive Bayes (NB) algoritmalarına tabi tutarak; f1-skor, kesinlik, hassasiyet ve doğruluk ölçevlerinden faydalanmışlardır. Bu ölçevler arasından doğruluk değeri 87% olan LR ölçevinin en uygun algoritma olduğuna karar verip Python dili kullanarak bir web sayfası oluşturmuşlardır. Buraya bir kullanıcı giriş ekranı eklemişler; siteye kayıt olmak isteyen kullanıcılardan yine Twitter kullanıcı adlarını kullanmalarını talep etmişlerdir. Siteye kayıt olan kullanıcının, Twitter platformunda adının geçtiği tweet'ler (kullanıcının yaptığı paylaşımlar) ve retweet'ler (alıntı paylaşımlar) sitenin veri tabanına kaydolmuş; böylece bu veriler, daha önce eğitildiği bahsedilen algoritma tarafından incelemeye tabi tutulabilmiştir. Bu sistem sayesinde ilgili kullanıcının paylaşımlarının hangilerinin Siber Zorbalık içerdiği, hangilerinin ise içermediği üzerine yorum getirebilen bir tablo elde edilmiştir (Dinçer, Kayaoğlu ve Safarlı, 2022).

2.3. Evaluating the Effectiveness of Different Machine Learning Approaches for Sentiment Classification

Iğdır Üniversitesi Mühendislik Fakültesi, Mekatronik Mühendisliği bölümünden Seda BAYAT ve Bilgisayar Mühendisliği bölümünden Gültekin IŞIK, "Evaluating the Effectiveness of Different Machine Learning Approaches for Sentiment Classification" (Duygu Sınıflandırmasında Farklı Makine Öğrenimi Yaklaşımlarının Etkinliğinin Değerlendirilmesi) adlı çalışmalarında; Amazon e-ticaret sitesindeki yorumları, geleneksel olarak ifade ettikleri Çok Katmanlı Algılayıcı (MLP – Multi Layer Perceptron), Naive Bayes ve Karar Ağacı (Decision Tree) gibi algoritmaların

yanı sıra daha modern olarak tanımladıkları, Transformer (Dönüştürücü) tipi bir algoritma olan BERT (Bidirectional Encoder Representations from Transformers - Transformatörlerden Çift Yönlü Kodlayıcı Gösterimleri) algoritmasının daha hafifletilmiş ve hızlandırılmış bir versiyonu olduğunu tarif ettikleri DistilBERT algoritması ile analiz etmeye odaklanmışlardır.

BAYAT ve IŞIK, çalışmalarında duygu analizinin sadece olumlu, olumsuz, nötr şeklinde kutuplar üzerinden yapılmadığı, aynı zamanda kızgınlık, mutluluk, üzüntü gibi farklı duygular açısından da gerçekleştirilebildiği olgusundan bahsetmişler; bu sayede müşteri geri bildirimleri, ürün yorumlamaları, sosyal medya gönderileri gibi veri elde edilen mecralardan gelecek duygu tespitleriyle kişilerin ihtiyaçlarına cevap verilebileceği, hatta işler arasındaki aciliyetin belirlenerek iş önceliğinin ve acil geri dönüşlerin sağlanabileceği şeklinde tespitlerde bulunmuşlardır. Öte yandan çalışmalarını "Coarse-grained sentiment analysis" (İri Taneli Duygu Analizi) yani daha yüzeysel ve geniş açıdan bakmayı sağlayan, sadece olumlu/olumsuz ayrımı üzerine yoğunlaşan bir analiz türü ile gerçekleştirmeye karar vermişlerdir. Bunu yapmak için, Github veya Hugging Face gibi platformlardan elde edilebilen "Amazon Reviews –Polarity (AR-P) Dataset" adlı hazır veri kümesinin, 200.000 adet örnekli bir alt kümesini kullanmışlardır. Veri kümesini üç alt kümeye daha, 70:15:15 oranlarıyla eğitim, doğrulama, test amaçlı olarak bölmüşler; ön işlemeler gerçekleştirerek DistilBERT, MLP, Naive Bayes ve Decision Tree sınıflandırıcılarını kullanmışlar ve modellerin performansını doğruluk, kesinlik, hatırlama ve F1 puanı ölçevleriyle karşılaştırmışlardır. DistilBERT, %96 civarlarındaki Doğruluk ve F1-Skoru değerleriyle daha geleneksel olan MLP, Karar Ağacı ve Naive Bayes algoritmalarını geride bırakmıştır. MLP algoritması da Karar Ağacı ve Naive Bayes algoritmalarına göre %85,06'lık Doğruluk oranı ile öne çıkmıştır. Çalışma sonucunda metin bazlı duygu analizinde transformer tipi algoritmaların başarılı olduğu fikrine ulaşılmıştır (Bayat ve Işık, 2023).

2.4. Twitter'da Duygu Analizi

Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nden Nagehan İLHAN ve Elektrik-Elektronik Mühendisliği Bölümü'nden Duygu SAĞALTICI, Harran Üniversitesi Mühendislik Dergisi'nde yayımlanan "Twitter'da

Duygu Analizi" adlı araştırma makalelerinde tıpkı DİNÇER, KAYAOĞLU ve SAFARLI'nın yaptığı gibi Twitter verileri üzerine yoğunlaşmışlardır. Onlar da tıpkı AYTEKİN ve BAYRAM'ın yaptığı gibi; veri hacminin gün geçtikçe büyümesi nedeniyle bu verinin analizinin zorluğuna ve sistematik bir yaklaşım gerektiğine dikkat çekmişlerdir. Çalışmalarında Twitter'den aldıkları 1.578.627 adet sınıflandırılmış tweet'i, anlamlarına göre 0 ve 1 olarak işaretlediklerini; Python'ın NLTK (Natural Language Toolkit – Doğal Dil Araç Seti) kütüphanesi ile simge ve noktalama işaretlerini temizlediklerini ve kelime köklerini bulduklarını, AYTEKİN ve BAYRAM'ın çalışmasındaki gibi "Jetonlaştırma" işlemi gerçekleştirdiklerini belirtmişlerdir. Takip eden süreçte "POS Tagger" adlı işlemle kelimeleri türlerine göre işaretlemişler, "uni-gram" ve "bi-gram" (N-gram türleri) şeklinde birer ya da ikişer olarak öbekledikleri kelimeleri eş anlamlılık bakımından da "Synset" adlı kütüphane ile incelemişler, SentiWordNet ile polarite hesaplamışlar, veri kümesini 66:33 oranlarında eğitim ve test amaçlı ayırarak ilgili SVM ve Naive Bayes makine öğrenmesi algoritmalarını uygulamışlardır. Uygulamalar sonucunda Naive Bayes ile %42'lik bir başarı elde ederken, uni-gram ve bi-gram yöntemli SVM algoritmasıyla %64'lük birer başarıya ulaşmışlardır. Buradan hareketle yaptıkları çalışma ile: ilgili veri kümesi ve uyguladıkları yöntemler göz önünde bulundurularak, ikili ve büyük verili bir metin sınıflandırmasında SVM'nin Naive Bayes yöntemine göre öne geçtiği sonucuna varmışlardır (İlhan ve Sağaltıcı, 2020).

2.5. Duygu Analizi İle Kişiye Özel İçerik Önerme

Düzce Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümünden Beyzanur BOSTANCI ve aynı üniversitenin Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü'nden Ahmet ALBAYRAK, "Veri Bilimi Dergisi'nde" yayımlanan "Duygu Analizi ile Kişiye Özel İçerik Önerme" adlı çalışmalarında; hem yukarıda bahsedilen diğer çalışmalardaki gibi Twitter hem de diğer çalışmalardan farklı olarak Facebook platformlarındaki kullanıcı yorumlarına, yine duygu analizi teknikleri üzerinden eğilmişlerdir. Yine diğer çalışmalardan farklı olarak, üniversite tercih döneminde olan kişilerin paylaşımlarını içeren kısıtlı bir veri kümesi üzerinde çalışmışlar; bu veri kümesinin seçilmesinin nedenini de "üniversite tercih dönemlerinde özel ve vakıf üniversitelerinin verdikleri reklamların sosyal medya

kullanıcı profilleri analiz edilerek kişiye özel içerik oluşturma amacıyla kullanılmak istenmesi" olarak açıklamışlardır.

Çalışmanın tanıtımından, okuyucuda; araştırmacıların bu çalışmaya özel üniversitelerin, özellikle ellerindeki mali güç nedeniyle kamu üniversitelerine göre daha kolay ve daha çok sayıda reklam verebilmeleri, böylece kamu üniversitelerine göre daha iyi tanıtım yapabilmeleri ve bu durumun da eşitliği bozduğu inançları üzerine yönlendikleri intibası oluşmaktadır. Çalışmayı önerirken, kamu üniversiteleri için eşitlik çözümünü, televizyon ve gazetelere göre daha uygun fiyatlı olduğunu belirttikleri sosyal medyadan tanıtım yapılması olarak belirtmişler; bunun için de iki aşamalı bir süreç ortaya koymuşlardır: 1) Facebook ve Twitter yorumlarının "iyimser", "karamsar", "mizahi", "üretken" ve "dışa dönük" şeklindeki kategorilerle tespit edilmesi, 2) Kullanıcı profilinin bu 5 kategori ile sınıflandırılması ve kullanıcıya uygun reklamın (içeriğin) hazırlanarak sunulması.

Tıpkı diğer çalışmalarda belirtildiği gibi bu çalışmada da ilgili mecralardan verinin elde edilmesi ve ön işleme aşamalarıyla süreç başlamıştır. İlgili veriler, Twitter ve Facebook platformlarında Geliştirici (Developer) hesaplarının açılarak ve API (Application Programming Interface – Uygulama Programlama Arabirimi) kullanılarak elde edilmiş, istenmeyen noktalama işaretleri, karakterler vs. ortadan kaldırılmış, kelime kökleri ortaya çıkarılmış, her kategori için 25'er anahtar kelime belirlenerek kelimelerin dokümanda / yorumda geçme sıklığına göre puanlanmaları ve veri kümesindeki her sözcüğün de ne kadar tekrar ettiğinin de TF-IDF (Term Frequency – Inverse Document Frequency / Terim Sıklığı – Ters Belge Sıklığı) yöntemiyle bulunarak ağırlıklarının belirlenmesi sağlanmıştır. Diğer çalışmalarda genelde izlenen yöntemden farklı olarak, olumlu/olumsuz gibi ayrımlar yerine yukarıda bahsedilen 5 kategori tercih edilmiş; en yüksek Doğruluk değerini %77,82'lik oranla "Dışadönük" tip kullanıcı yorumlarının elde ettiği tespit edilmiştir.

Çalışmanın sonunda ise sürecin bahsedilen ikinci aşaması olan "İçerik Sunma" gerçekleştirilmiş; araştırmacılar, bahsedilen 5 kategori için de kendi üniversiteleri olan Düzce Üniversitesi'ni tanıtan birer afiş hazırlamışlardır (Bostancı ve Albayrak, 2021).

Bölüm 3

Metodoloji

3.1. Veri kümesinin Elde Edilmesi

Internet üzerinden erişilen çeşitli mecralarda araştırma makalesi veya lisansüstü tezi gibi pek çok belge, dolayısıyla çalışma yer almaktadır. Yukarıdaki ilgili bölümde 5 tanesi özetlenen bu çalışmalar, tıpkı bu çalışmamızda olduğu gibi Duygu Analizi üzerine konumlandırılmıştır. Duygu analizi çalışmaları, makine öğrenmesi veya derin öğrenme gibi yapay zekâ sistemlerinin eğitilmesi ve bu sistemlerin önlerine gelen gerçek verileri sınıflandırmaları gibi işlemler sonucunda gerçekleştirilir. Söz konusu sistemlerin eğitilebilmesi için belli bir sayıda veriye ihtiyaç duyulacağı açıktır. Bu şekildeki bir veri kümesi, ancak belirli kaynaklardan elde edilebilir. Duygu analizi üzerine yapılacak bir çalışmanın veri kümesi, araştırmacılar tarafından aşağıdaki iki yöntemle elde edilebilir:

1. Hazır veri kümelerinin Kaggle gibi çeşitli sitelerden veya kurumlardan talep edilerek alınmasıyla,

2. Araştırmacının kendi veri kümesini web kazıma gibi yöntemlerle elde etmesiyle.

Yukarıda da örneği verilen çalışmaların bazılarında hem Kaggle gibi sitelerden hazır alınmış veri kümelerinin, üzerlerinde gerçekleştirilen işlemler sonrasında kullanıldığı hem de Twitter API gibi araçlarla araştırmacıların kendileri tarafından verilerin elde edilerek düzenlendiği ve birer veri kümesine dönüştürüldüğü görülmüştür.

Bu çalışmamız, bahsedilen ikinci yöntemle, yani web kazıma yöntemi kullanılarak veri kümesinin elde edilmesine dayalı bir şekilde gerçekleştirilmiştir. Yalnız burada çok önemli bir fark bulunmaktadır. Web kazıma yöntemi için herhangi bir araç satın alınmamış; araştırmacının kendi araştırması ile bulduğu bir Youtube videosu, çalışmanın başlangıcını oluşturmuştur.

3.2. Web Kazıma Kavramı, John Watson ROONEY ve Web Kazıma İşlemleri

3.2.1. Web Kazıma Kavramı

Bir internet sitesindeki bilginin bir bilgisayarda da kopyasının oluşturularak elde edilmesinin akla gelen ilk yolu; ilgili sitede herhangi bir engel bulunmuyorsa, almak istenen metnin fare ile tıklanarak seçilip kopyalanıp bilgisayardaki ilgili uygun alana yapıştırılması olacaktır. Fakat bazı durumlarda, gerekli bilginin aynı sayfa içindeki uzun bir metinden elde edilmesi, doğrudan seçilmesi, kopyalanıp yapıştırılması mümkün olmayabilir; bunun için çok sayıda sayfanın ayrı ayrı açılması ve bu sayfalar içindeki belli alanların seçilerek kopyalanıp hedef ortama yapıştırılması ve her sayfa için bu işlemlerin tekrarlanması gerekir. Bu da tahmin edilebileceği üzere çok zaman ve emek isteyen bir süreçtir. İşte web kazıma (Web Scraping) işlemi burada devreye girmektedir.[2]

Web kazıma, yukarıda anlatıldığı gibi, bilginin her seferinde elle kopyalanıp hedef ortama yapıştırılması yerine, ilgili siteden tüm veya gerekli verileri çıkarma işlemine verilen isimdir. Veri, bir web sitesindeki en az bir sayfadan indirilebilir; bu da sitedeki içerik türleri tanınarak yapılır ve sadece kullanıcının belirlediği içerik alınır, saklanır. Bu teknik, farklı yapılara sahip web sitelerinden hem yapılandırılmış hem de yapılandırılmamış verilerin elde edilmesini sağlamaktadır (Poongodai ve Suhasini, 2019).

Web kazıma yönteminin arama motoru kullanmaktan en büyük farkı, çok daha fazla bilgiye erişilmesini sağlamasıdır. Örneğin bir Google arama motoru ile belirli bir sitede arama yapılırsa; arama yapan kullanıcı, önüne gelecek reklamlara maruz kalabilir, ayrıca istenen nokta atışı bilgilere erişilmesi dolaylı yolla gerçekleşebilir. Arama motoru, sadece ilgili sitelerin içerikleri hakkında genel bir bilgi verir ve kullanıcıyı istediği hedefe yönlendirir; ancak amaca uygun bir şekilde hazırlanmış bir web kazıyıcı ile kullanıcı, çeşitli web sitelerindeki aramak istediği konular hakkında

2 Araştırmacının kendi gözlemlerine dayalı ifadeleridir.

net detaylara erişebilir; örneğin belirli bir bölgedeki otellerin konaklama maliyetleri ve rezervasyon için en iyi zamana göre en uygun otelleri tespit edebilir. Web kazıma araçlarının bu kadar verimli olması; kullanıcının, çeşitli kütüphanelerin yardımıyla yazıp çalıştırdığı bir kod dizilimi sayesinde; gerekli bilginin alınması için ilgili sitede nereden başlanıp nerede durulması, hangi bilgilerin ne kadarının, nasıl alınıp nereye kaydedilmesi gerektiğini ilgili programa tanıtabilmesi ve programın kullanıcının bu isteklerini otomatik olarak gerçekleştirebilmesinden ileri gelmektedir (Poongodai ve Suhasini, 2019).

3.2.2. Yapılandırılmış ve Yapılandırılmamış Veri Kavramları

Genellikle nicel (sayılabilen/sayısal) veri olarak kategorize edilen **yapılandırılmış veriler**, makine öğrenimi algoritmaları tarafından kolayca çözülebilir. Farklı veri türlerinin ve bunların nasıl işlendiğinin derinlemesine anlaşılmasını gerektirmez; bu yüzden farklı kullanıcılar, verilere kolayca erişebilirler ve bunları yorumlayabilirler. Yapılandırılmış verileri kullanmak ve analiz etmek için daha fazla araç mevcuttur. Fakat önceden tanımlanmış bir yapıya sahip veriler yalnızca amaçlanan amaç için kullanılabilir, bu da esnekliği ve kullanılabilirliği sınırlar. Veri gereksinimlerindeki değişiklikler tüm yapılandırılmış verilerin güncellenmesini gerektirir ve bu da büyük miktarda zaman ve kaynak harcamasına yol açar. Yapılandırılmış verilere; tarihler, isimler, adresler, kredi kartı numaraları; müşteri davranış kalıpları ve eğilimleri üzerine çalışılan CRM (Müşteri İlişkileri Yönetimi) verileri, otel ve bilet rezervasyon verileri, muhasebe kayıtları örnek verilebilir.

Genellikle nitel veri olarak sınıflandırılan **yapılandırılmamış veriler**, geleneksel veri araçları ve yöntemleriyle işlenemez ve analiz edilemez. Araştırmalar, yapılandırılmamış verilerin tüm kurumsal verilerin %80'inden fazlasını oluşturduğunu gösterdiğinden; işletmelerin büyük çoğunluğu, bu verilerin yönetimine öncelik vermektedir. Saklanan yapılandırılmamış veriler, ihtiyaç duyulana kadar tanımsız kalır; veri tabanındaki dosya formatları artar, veri havuzu genişler ve yalnızca ihtiyaç duyulan verinin analizine gerek duyulur. Bir ön tanımlama gerekmemesi; hızlı ve kolay toplanılmasını sağlar. Fakat biçimlendirilmemiş yapısı nedeniyle, hazırlanması ve analizi için uzmanlık gerekir. Verileri nasıl kullanılacağını tam olarak bilmeyen kişiler için bu bir sorundur.

Yapılandırılmamış verilere mobil etkinlikler, Nesnelerin İnterneti (IoT) verileri, sosyal medya gönderileri veya sitelerdeki kullanıcı yorumları gibi sözel bilgiler, bir işyerindeki kurallar veya bu işyerinde çalışan kişilerin notları gibi veriler, yani aslında çoğunlukla "metinler", "sözel ifadeler" örnek verilebilir. Bu veriler, doğrudan doğruya sayısal olarak düşünülemeyen ve üzerinde çalışılması gereken veriler olacaktır.[3]

3.2.3. John Watson ROONEY ve Web Kazıma Metodu

John Watson ROONEY, Birleşik Krallık merkezli bir YouTuber'dır. "YouTuber" kavramı, Wikipedia.org'ta "Çevrimiçi video paylaşım sitesi YouTube'a videolar yükleyen veya video oluşturan, genellikle kişisel YouTube kanallarına gönderiler gönderen bir tür sosyal medya etkileyicisi" olarak tanımlanmaktadır. Terimin ilk olarak 2006 yılında İngilizce dilinde, Time Dergisinin "Yılın Kişisi" sayısında kullanıldığı bilgisi, sitede yazmaktadır. YouTuber'lar Google AdSense adlı mecradan, ek olarak Patreon gibi platformları kullanarak satış ortağı bağlantıları, satış ve 3. taraf üyelikler yoluyla gelir elde edebilmektedirler. Popüler kanallar, videolara dahil olmak için para ödeyen kurumsal sponsorlar kazanmakta olup; 2018 yılında Walmart, Nordstrom gibi firmaların, "Influencer" adıyla da bilinen ve diğer kişileri etkileyen bu kişilerle temasa geçtiği bilgisi, yine sitede görülmektedir.[4]

ROONEY'in ilgili YouTube kanalında, programcılık ve bilişim üzerine pek çok videosu yer almaktadır. Bunlar arasında yer alan, 4 Kasım 2020 tarihli "How I Scrape Amazon Reviews using Python, Requests & BeautifulSoup" *(Python ortamındaki Requests ve BeautifulSoup kullanarak Amazon Ürün İncelemelerini Nasıl Kazıyorum?)*[5] adlı videosu, çalışma için gerekli olacak web kazıma kodlarının nasıl yazılacağı hakkında araştırmacıya yol göstermiştir.

---

3 Ibm.com; "STRUCTURED VS. UNSTRUCTURED DATA: WHAT'S THE DIFFERENCE?" https://www.ibm.com/blog/structured-vs-unstructured-data/; Erişim: 27.01.2024

4 Wikipedia.org; "YOUTUBER"; https://en.wikipedia.org/wiki/YouTuber; Erişim: 26.01.2024

5 YouTube; ROONEY, J.W. (2020). HOW I SCRAPE AMAZON REVIEWS USING PYTHON, REQUESTS & BEAUTIFULSOUP; https://www.youtube.com/watch?v=DIT8rwyPEns; Erişim: 27.01.2024

Python programlama dili, ilgili sitesinde şöyle tanımlanmaktadır:

"Python; dinamik anlambilime sahip, yorumlanmış, nesne yönelimli, üst düzey bir programlama dilidir."[6]

Python'un basit ve öğrenmesi kolay sözdiziminin, kendisini okunabilir kıldığı; dolayısıyla program bakım maliyetini azalttığı, modülerliği ve kodun yeniden kullanımını teşvik eden modülleri ve paketleri desteklediği belirtilmektedir. Yorumlayıcısının ve kapsamlı standart kitaplığının, tüm önemli platformlar için ücretsiz ve serbestçe dağıtılabilir olduğu bilgisi, yine sitede yer almaktadır. Pek çok programlama dilinin tersine, Python'da bir derleme adımı olmadığı; bu yüzden de düzenleme, test ve hata ayıklama süreçlerinin daha kolay ve hızlı olduğu gözlemlenebilmektedir.

Requests, Python ortamında kullanılan bir kütüphanenin adıdır. İlgili sitesinde kütüphane ile ilgili kısa ve öz bir tanımlama bulunmaktadır:

"Requests basit ama zarif (elegant) bir HTTP kitaplığıdır."[7]

Requests'in, web sitelerine HTTP 1.1 protokolü kullanarak, ek komut satırları eklemeden, basit bir şekilde istekler gönderebildiği; yine sitenin dokümantasyonunda yer alan bir bilgidir.

HTTP (The Hypertext Transfer Protocol / Köprü Metni Aktarım Protokolü) kavramı, Wikipedia.org sitesinde *"dağıtılmış, işbirlikçi, hiper ortam bilgi sistemleri için İnternet protokol paketi modelindeki bir uygulama katmanı protokolü"* olarak tanımlanmaktadır. HTTP, kullanıcıların bilgisayarlarıyla aracılığıyla, Internet olarak da ifade edilen "World Wide Web" ortamındaki diğer kaynaklara erişimlerini sağlayan köprüleri içerir; bu nedenle de ilgili iletişimimizin temelini oluşturmaktadır.[8]

6 Python.com; "WHAT IS PYTHON? EXECUTIVE SUMMARY"; https://www.python.org/doc/essays/blurb/; Erişim: 27.01.2024

7 Pypi.org; "REQUESTS"; https://pypi.org/project/requests/; Erişim: 27.01.2024

8 Wikipedia.org; "HTTP"; https://en.wikipedia.org/wiki/HTTP; Erişim: 31.01.2024

Beautiful Soup kütüphanesi, ilgili sitesinde şu şekilde tanımlanmaktadır:

"Beautiful Soup; HTML ve XML dosyalarından veri çekmeye yönelik bir Python kütüphanesidir."

Beautiful Soup'un yine ilgili sitesinde; compiler adı verilen kod düzenleyicilerinde, ayrıştırma ağaçlarında gezilmesini ve arama yapılmasını sağladığı; programcılara çalışmalarından genel anlamda saatlerce, hatta günlerce tasarruf sağladığı şeklinde ifadeler bulunmaktadır.[9]

Splash, kendisine ait dokümantasyon sitesinde şöyle tanımlanmaktadır:[10]

"Splash bir javascript oluşturma hizmetidir. HTTP API'ye sahip, hafif bir web tarayıcısıdır."

Splash, Docker Desktop adlı bir program dahilinde çalıştırılmaktadır.

Docker Desktop, konteynerli (containerized) adı verilen uygulamaların çalıştırılmasına olanak tanıyan bir uygulamadır. Başka yazılımların bilgisayar üzerinden kendi dahilinde yürütülmesini sağlamaktadır. Bu tanımlamaya uygun bir şekilde Docker Desktop, önce Microsoft Windows ortamına yüklenmekte; sonrasında da Splash adlı uygulama yüklenmektedir. Docker Desktop açıldıktan sonra (bir konteyner gibi / *containerized*) ev sahipliği yaptığı Splash yazılımı çalıştırılmaktadır.

Amazon sitesi; pek çok diğer web sitesi gibi kendisini sistematik saldırılardan, bot adını verdiğimiz robotlardan koruyabilmek için savunma mekanizmaları geliştirmiştir. Bu savunma mekanizmalarının sonucunda kullanıcının uğrayabileceği mağduriyetler bulunmaktadır. Eğer kullanıcı, siteden otomatik bir şekilde, ortalama bir insanın web tarayıcılarıyla, klavye ve fare ile site adları girmek, ilgili bağlantılara tıklamak gibi normal bir hızda yaptığından daha hızlı bir şekilde veri alırsa; Amazon sitesi bu kişinin bilgisayarını tanımlayan IP numarasına blokaj getirebilmektedir. Bu durum, kişinin

9 Crummy.com; "BEAUTIFUL SOUP DOCUMENTATION"; https://www.crummy.com/software/BeautifulSoup/bs4/doc/; Erişim: 26.01.2024

10 Splash.readthedocs.io; SPLASH – "A JAVASCRIPT RENDERING SERVICE";

https://splash.readthedocs.io/en/stable/; Erişim: 26.01.2024

siteye erişimini kısıtlayabilecek, belki de tamamen engelleyebilecektir. Bu tehlikeye karşı Docker Desktop ve Splash kullanmak bir yöntemdir. Splash, Python'da yazılan kodu başka bir internet adresine yönlendirmektedir. Böylece otomatik bir şekilde Beautiful Soup ve Requests kütüphaneleriyle çekilen verilerin sonucu olarak kullanıcının, tamamen veri bilimi amacıyla, siteye bir saldırı veya başka kötü bir niyet taşımadan gerçekleştirdiği eylem sonucu cezalandırılması engellenebilmektedir. Bu çalışmada da ilgili metot kullanılmış; ROONEY'in aşağıdaki bölümde açıklayacağımız Python kod silsilesi ile çeşitli ürünlerle ilgili kullanıcı yorumları elde edilebilmiş; bunun sonucunda da kodun çalıştırıldığı bilgisayar, Amazon sitesi tarafından bloke edilmemiştir.

ROONEY, kodun nasıl yazılacağını anlamak için öncelikle Amazon sitesindeki internet adreslerinin nasıl oluşturulduğuna odaklanmıştır. Sitedeki binlerce ürün arasından rastgele seçilen bir tanesinin ilgili sayfasına girilerek kullanıcı yorumlarının olduğu bölüme erişildiğinde, web tarayıcısının adres satırında şu şekilde bir örnek adres oluşmaktadır:

https://www.amazon.com/product-reviews/B09TZWLFLY/ref=acr_dp_hist_5?ie=UTF8&filterByStar=five_star&reviewerType=all_reviews#reviews-filter-bar

Bu örneğimizdeki "B09TZWLFLY" ifadesi, Amazon.com sitesinin her ürün için oluşturduğu bir ID numarasından sadece biridir. Hatta sitenin arama motoruna bu ifade yazılırsa da ilgili ürüne ulaşılabilmektedir. "UTF8", karakter tipini, "filterByStar=five_star" ifadesi, sadece 5 yıldızlı yorumların seçildiğini *(ürünler; 1 en düşük, 5 en yüksek olacak şekilde 5 farklı yıldız sayısı ile puanlanabilmektedir),* "reviewerType=all_reviews" ifadesi ise bu filtreler içindeki tüm yorumların çağrıldığını göstermektedir.

İkinci sayfaya geçildiğinde ise adres şöyle değişmektedir:

https://www.amazon.com/product-reviews/B09TZWLFLY/ref=cm_cr_arp_d_paging_btm_next_2?ie=UTF8&filterByStar=five_star&reviewerType=all_reviews&pageNumber=2#reviews-filter-bar

Buradaki farklılıklar arasından şu ifade dikkati çekmektedir: “&pageNumber=2#reviews-filter-bar”. Burada sayfa numarasının tanımlandığı görülmektedir. Buradan da ilgili ürünün tüm yorumlarının elde edilebilmesi için ilgili sayfa sayısı yerine “x” değerinin verilebileceği ve başka bir fonksiyon içindeki bir “for” döngüsü ile tüm sayfaların çekilebileceği fikri oluşmaktadır.

ROONEY, sonrasında ilgili yorum bölümü açıkken sitenin HTML Java Script kodlarını incelemiş ve hangi kod bloğundaki hangi alanın, gerekli Python kodunu etkileyeceğini tespit etmiştir. Bir for döngüsü kurarak ilgili kod bloğunda şu parametreleri indirecek kodları yazmıştır: “product”, “title”, “rating” ve “body”. Product, ilgili ürünün uzun adıdır; yani örnekteki adreste “product”, şudur: *“Dacoity Gaming Keyboard, 104 Keys All-Metal Panel, Rainbow LED Backlit Quiet Computer Keyboard, Wrist Rest, Multimedia Keys, Anti-ghosting Keys, Waterproof Light Up USB Wired Keyboard for PC Mac Xbox”*. “Title” ise ilgili kullanıcının yoruma başlamadan önce koyduğu başlıktır, söz gelimi bir olumlu yorumun Title’ı şu olacaktır: *“Better than expected!” (Umduğumdan daha iyi!).* “Rating” ise kullanıcının, 1’den 5’e kadar verebildiği puandır. İlgili kazıma sonucunda 5 üzerinden 5 puanlık bir yorum için “Rating” şöyle gelecektir: “*5 out of 5 stars” (5 üzerinden 5)*. ROONEY, for döngüsünü çalıştırırken, “5 out of 5 stars” yerine sadece “5” ifadesi gelmesi için 'out of 5 stars' ifadesini sildirmekte, kalan “5” ifadesini de float’a (kesirli sayıya) çevirmektedir. Son olarak “body” bölümü de bu çalışmayı önemle etkileyecek “yorum” metnidir; yani kullanıcının başlık atıp, ilgili yıldızlı puanlamayı yaptıktan sonra yazdığı “yorum metnidir”. Python’ın “strip” komutuyla bu yorum metnindeki gereksiz boşlukların ve her yorumun başında görülen “Amazon.com: Customer reviews: ” ifadesinin de silindiği, kodda görülmektedir.

ROONEY, kodun işleyişinin nerede biteceğine; ancak ürün yorumlarının son sayfasına ulaşıldığında görülebilen ve bu sayfadaki Java Script kodunun ilgili etiketi altında yer alan *“a-disabled a-last”* ifadesini tespit ederek karar vermiştir. Buna göre kodda ikinci bir “for” döngüsü açmış, 1 ile 999. sayfa arasında döngüyü çalıştırarak, ilgili “a-disabled a-last” ifadesi bulunmadığı sürece döngünün, dolayısıyla da yorumların alınmaya devam edilmesi komutunu oluşturmuştur.

Kodun son bloğunda da elde edilen tüm bu bilgilerin, .xlsx formatında bir Excel dosyasına aktarılması söz konusudur. Her bir ürün için bu kod çalıştırıldığında, ürüne ait yapılmış olumlu ya da olumsuz tüm yorumların tek bir excel dosyasında toplanması beklenmektedir.

3.3. ROONEY'in Metodunun Uygulanması ve Karşılaşılan Zorluklar

Araştırmacı; ROONEY'in 4 Kasım 2020 tarihli ilgili Youtube videosuna, 2023 yılının Mayıs ayında ilk defa erişmiştir. Araştırmacının videoya erişim amacı; "Yazılım Mühendisliği" ana bilim dalında tezsiz yüksek lisans eğitimini gördüğü sırada aldığı "Metin Madenciliği" dersinin projesinde kullanacağı, görece daha küçük bir veri kümesini, kendince oluşturabilme isteğidir. Videoda anlatılan yöntemler ve tanıtılan kodlar ile tıpkı ROONEY'in yaptığı gibi Amazon.com sitesinden web kazıma işlemlerini başarıyla gerçekleştirmiştir. Bu nedenle, ikinci yarıyıl sonrasında, 2023'ün yaz aylarına gelindiğinde ise araştırmacı; eğitimindeki son dönem olan üçüncü yarıyıl henüz başlamadan, ilgili yüksek lisans programının bitirme projesinde kullanacağı çok daha büyük bir veri kümesini elde edebilmek için aynı yöntemi, dolayısıyla aynı kodları kullanabileceğine karar verir.

3.3.1. Zorluk 1: Amazon.com sitesinin yapısının değişmesi

Üçüncü yarıyıl başlamadan önce yapılan denemelerde; artık ilgili amazon sitesinde en fazla 10 sayfalık yorumlara ulaşılabildiği gerçeğiyle karşılaşılmıştır. 10. sayfa her zaman yorumların son sayfası haline gelmiştir. Videoya ilk erişim sağlandığı Mayıs 2023 tarihlerinde onlarca, hatta var ise yüzlerce sayfalık yoruma ulaşılabilirken; bu durum artık değişmiştir. Site, yapısı gereği sayfa başına en fazla 10 adet yorum görüntülemektedir. Buradan hareketle site, yıldız ayrımsız tüm yorumların çağrılmasıyla; ürün başına en fazla 100 adet (10 x 10 = 100) yorumu kullanıcıya göstermektedir. Araştırmacı, bu sorunu aşmak için şu çözümü getirmiştir: 1, 2, 3, 4 ve 5 yıldızlı yorumları ayrı ayrı çağırmak ve bunların tüm yorumlarını ayrı ayrı kazımak. Araştırmacı, ROONEY'in ilgili kod silsilesini, tek bir kod içinde 5 kere tekrarlatmış; 1, 2, 3, 4 ve 5 yıldızlı olacak şekilde art arda gelen kodları, sadece ürün ID'lerini

değiştirerek, her ürün için tek kodda çalıştırmış; her ürünün her yıldız sayısına göre 5 farklı Excel dosyası elde etmiştir. Bu Excel dosyalarını da elle birleştirerek, ürün başına elde edebildiği azami yorum sayısını, 100 yerine 500'e çıkarmayı başarmıştır. Ayrıca veri kümesinin elde edilmeye başlandığı ilk zamanlarda, en çok yorum yapılan elektronik aletlerin fotoğraf makineleri olduğu gözlemlenmiş; bu nedenle de esasen sadece fotoğraf makinelerinin yorumlarının elde edilmesi planlanmıştır. Fakat öne çıkan bu sorun, bu planın değiştirilmesini mecbur hale getirmiş; farklı ürünlerin de yorumlarının alınması yoluna gidilmiştir. Amazon.com sitesinde çok farklı tipte ürün satılıyor olmasına rağmen; yorumlardaki bağlamın, elektronik aletler dışındaki ürün türlerinin de yorumlarının alınmasıyla kaybolabileceği göz önünde bulundurularak; sadece USB bellek, bilgisayar parçaları, kulaklıklar, webcam ve hoparlörler gibi nihai kullanıcının yoğunlukla talep ettiği diğer elektronik ürünlerin yorumlarının elde edilmesi yoluna gidilmiştir.

3.3.2. Zorluk 2: Amazon.com yorumlarındaki beklenmeyen ifadeler

Araştırmacının tek tek yazmadan ya da yorumu seçip kopyala-yapıştır yapmadan; Python kodlarıyla web kazıma yöntemini kullanarak elde ettiği yorumlar arasında, beklenmeyen ifadeler de yakalanmıştır. Bazı satırlarda *"The media could not be loaded."* ifadesinin, bazılarında ise şu şekilde uzun bir metnin yer aldığı fark edilmiştir: *"Video Player is loading. Play Video Play Mute Current Time 0:00/ Duration 0:00Loaded: 0%Stream Type LIVE Seek to live, currently behind live LIVE Remaining Time -0:00 1xPlayback Rate Chapters Chapters Descriptions descriptions off, selected Captions captions and subtitles off, selected Audio Track Fullscreen This is a modal window."*. Söz konusu ifadelerin, incelemeye alınan ürünler arasında olan fotoğraf makineleri veya hafıza kartlarından geldiği anlaşılmış, dolayısıyla en başlarda yorumların normal birer parçası olduğu düşünülmüş; fakat ürünlerin birbirinden ilgisiz yanlarına değinilen yorumlarda yine aynı şekilde geldiği; üstelik ilgisiz boşluklar bıraktığı görülerek bu görüşten vazgeçilmiştir.

Söz konusu hatalar, çeviri yapılırken fark edilmiş; Microsoft Excel'in (ve Libre Office'in) "Bul ve Değiştir" komutlarıyla ilgili ifadeler silinmiştir. Bu ifadelerden geriye kalan, fakat yine "Bul ve Değiştir" komutuyla silinemeyen boşluklar, Excel'in

ilgili şu komut satırıyla temizlenmiştir: “=KIRP (TEMİZ (‘Üzerinde işlem yapılacak Excel hücresi’))” *(İngilizce Excel’de: “=TRIM (CLEAN (‘Üzerinde işlem yapılacak Excel hücresi’))”)*.

3.3.3. Zorluk 3: Amazon.com yorumlarındaki dil farklılığı

Araştırmacı, ROONEY’in yaptığından farklı olarak Amazon.com sitesinden veri elde etmiştir. ROONEY, Birleşik Krallık kökenli bir kişidir ve Amazon.co.uk sitesini kullanmaktadır. Amazon.com ise şirketin esas kökeni olan Amerika Birleşik Devletleri (ABD) merkezlidir. Amazon.com ile Amazon.co.uk sitelerinin yapıları ve dolayısıyla kodları, neredeyse birbirinin aynısıdır; dolayısıyla aynı kodlarla ABD sitesinden de aynı şekilde veri elde edilebilmiştir.

Amazon.com sitesinden sadece ABD’de yaşayanlar değil; Dünyanın farklı pek çok ülkesinden, dolayısıyla milletinden kişiler alışveriş yapabilmekte ve siteye kendi dilleriyle yorum yazabilmektedir. Site, ABD dışından yazılmış yorumları gösterirken, kendi yapay zekâ altyapısı sayesinde, kullanıcıya otomatik çeviri imkânı da sunmaktadır. Kullanıcı, sunulan çeviri bağlantısına tıkladığında, yorum otomatik olarak çevrilebilmektedir. Yorumlar web kazıma yöntemiyle elde edildiğinde ise sadece ABD merkezli yorumların çekilebildiği görülmektedir.

Wikipedia.org’a göre Amerika Birleşik Devletleri’nin federal düzeyde bir resmi dili olmasa da 32 eyalet ve 5 bölge bazında resmi dili İngilizcedir. Ayrıca İngilizce dili, gayrı resmi bir şekilde milli dil olarak kabul edilmektedir. Fakat ABD’de 245 milyon civarında olduğu belirtilen İngilizce konuşan kişi sayısının yanında (%78,5’lik bir oranla), 41,3 milyon civarında (%13,2 oranında) İspanyolca konuşan kişi olduğu bilgisi de sitede ayrıca yer almaktadır.[11] Bu bilgi sayesinde, “her 10 ABD vatandaşından en az birinin, İngilizcenin yanında *ya da belki de yerine* İspanyolca konuştuğu” sonucuna varılabilir. Bu gerçek, ilgili yorumlara da yansımıştır. Tüm ürünlerin yorumları tek bir Excel dosyasında birleştirildiğinde görülmüştür ki: İlgili yorumların hatırı sayılır bir kısmı, tamamen İspanyolca diliyle yazılmış durumdadır.

[11] Wikipedia.org; "LANGUAGES OF THE UNITED STATES"; https://en.wikipedia.org/wiki/Languages_of_the_United_States; Erişim: 27.01.2024

Elde edilen yorumların, Google'ın çeviri ortamına Excel dosyasının yüklenerek otomatik bir şekilde çevrilmesi gerek teknik altyapı gerekse araştırmacının otomatik çeviri sonrasında metinlerdeki potansiyel bağlam kaybı endişesinden dolayı gerçekleştirilememiştir. Bu nedenle araştırmacı, yüzlerce satırlık metni, Google çeviri sayfasına satır satır yapıştırarak ve kendi İspanyolca bilgisini kullanarak "elle" çevirmiştir. Çalışmada Excel dosyalarında işlem yapılabilmesi için, araştırmacı ev bilgisayarında Libre Office adlı açık kaynaklı yazılımı, çalıştığı iş yerinde ise Microsoft Excel yazılımını kullanmıştır.

3.4. Bazı Terimler

Çalışmada yapılan ön işleme adımları; sonrasında da makine öğrenmesi ve derin öğrenme sınıflandırma algoritmaları kullanılarak yapılan işlemler öncesinde; anlatımlar içinde yer alacak bazı bilgi ve tanımlamaların tanıtılması yerinde olacaktır.

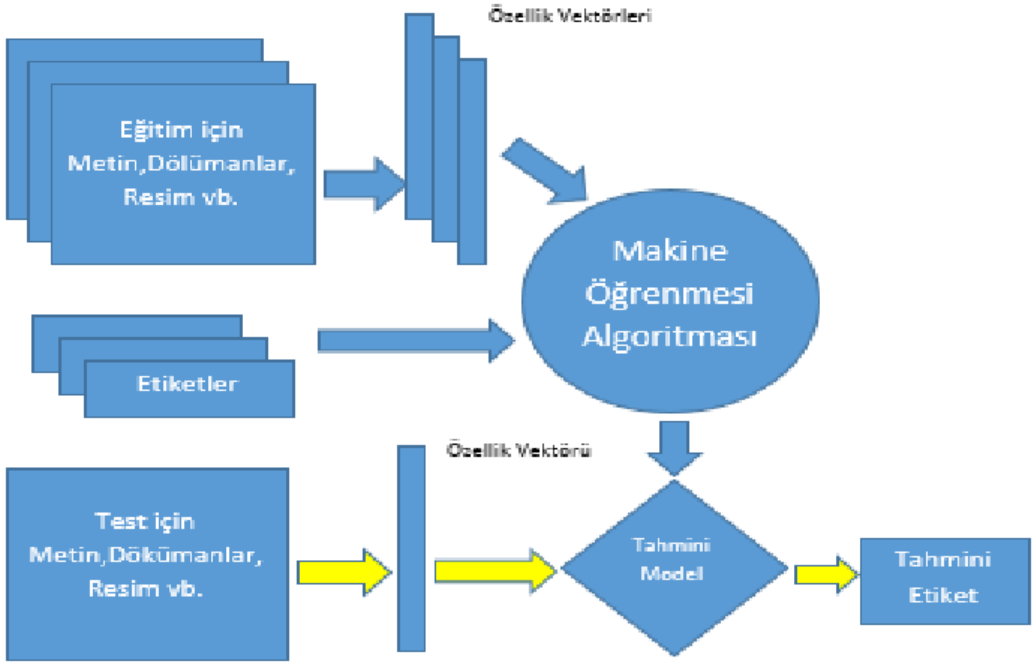
3.4.1. Sınıflandırma

Sınıflandırma dendiğinde akla gelmesi gereken şudur: Bir görüntünün/sesin/metnin, yani bir tür verinin, çeşitli örneklemler (instance) üzerinden özelliklerinin (veriyi tanımlayan farklı özellikler; öznitelik – feature) belirlenmesi ve bunların incelenerek bu verinin önceden tanımlanan sınıflara (etiket – label) atanmasıdır. Bu işlemin yapılabilmesi için özelliklerin ve atanacak sınıfın önceden, net bir şekilde belirlenmesi çok önemlidir (Sabancı, 2016). Sınıflandırmanın zihinde daha iyi şekillendirilmesi için bir grup öğrencinin ya da askerin; özelliklerine göre çeşitli görevlere seçilmesi düşünülebilir. Söz gelimi Türk Ordusu'nda uzun boylu ve görece yapılı denebilecek kişilerin; muhafız alayı ya da tören mangasına seçildiği; daha kısa boylu ve/veya kilolu kişilerin daha yönetimsel ya da ofis ortamlarında yapılması gereken işlere alındığı, Türkiye Cumhuriyet'inde zorunlu askerlik hizmetini yapmış kişilerce gözlemlenebilecek bir durumdur. Burada ele alınan; boy-kilo parametresi, yukarıda bahsettiğimiz *feature* kavramına, askerlerin her biri birer *örneklem'e*, yapması için seçildiği işler ise birer *label*'a örnek verilebilir.[12]

12 Araştırmacının kişisel gözlemlerine dayalı bir örnektir.

3.4.2. Denetimli Öğrenme

Denetimli Öğrenme (Supervised Learning, Danışmanlı Öğrenme), sistemin bir danışman/denetim mekanizması ile durumu öğrenmesine verilen addır. Söz konusu danışman, ilgili sisteme öğrenmesi istenen konuyla ilgili örnekleri Girdi / Çıktı olarak verir. Giriş ve çıkış değerlerinin eşleştiği örnekler baz alınarak bir fonksiyon öğrenilir ve bir hipotez üretilir. Gerçekleştirilen bir eğitim süreci sayesinde oluşturulan modelden, sistemin test süreçlerinde bir sınıflandırma yapması beklenir. Böylece öğrenilmesi hedeflenen girdilerin belirli sınıflara, dolayısıyla hedeflere yönlendirilmesi mümkün olmaktadır. Destek Vektör Makinesi (Support Vector Machine), Yapay Sinir Ağları (Artificial Neural Network), Naive Bayes, k-En Yakın Komşu (k-Nearest Neighbor) ve Karar Ağaçları (Decision Trees); Denetimli Öğrenme için verilebilecek uygun algoritma örnekleridir. Bu yöntem ile tek (single-label) veya çok etiketli (multi-label) sınıflandırmalar yapılması mümkün olmaktadır (Bilgin ve Şentürk, 2019).



Şekil 3.1 – Denetimli Öğrenme (Bilgin ve Şentürk, 2019)

3.4.3. Denetimsiz Öğrenme

Denetimsiz öğrenmede ise belirli bir hedef tanımlanmamaktadır. İlgili sistemin; girdileri ve girdilerin sağladığı bilgileri, bu girdilerden gelen verilerinin boyut ya da miktarlarının içsel azaltımları sayesinde ilişkilendirmesi söz konusudur. Aslında girdilerden gelen veriler arasında kurulmaya çalışılan korelasyonlar, ilgili öğrenmeyi sağlamaktadır; bir danışan sistemi olmadan ilgili verilerdeki kalıp veya özellikler bulunmaya çalışılır. Rekabet ve iş birliği ilkeleri kullanan bu sistem, sonlandırılması gereken bir süreci içerir. Bu sonlandırma kriteri ortaya konmadığı sürece; ilgili örüntünün yapısı ne olursa olsun, öğrenme devam edecektir. Bu tür özellikleri nedeniyle denetimsiz öğrenme için; süreç sırasında koşulları algılayan, temkinli hareket eden ve bu yüzden de yavaş bir süreç yürüten bir metot olduğu yorumu yapılabilir (Du ve Swamy, 2014).

3.4.4. Sürekli ve Süreksiz Değişkenler

Sürekli değişkenler, bir aralıkta tanımlanan; süreksiz değişkenler ise belli noktalarda duran değişkenler olarak çok kabaca tanımlanabilir. Örneğin; emlak piyasasındaki taşınmazların fiyatlarının tahminlemesi bir sürekli değişkenler durumudur; çünkü bu fiyatlar belirli bir aralıkta yukarı aşağı yönlü, sürekli değişebilir. Fakat bir kesedeki toplar sahip oldukları renklere göre sınıflandırılıp ayrı keselere alınıyor, sonra da aynı renkler arasından bir renge sahip başka bir top daha getirilip, hangi keseye dahil edileceği sorgulanıyor ise burada bir değişkenlik söz konusu değildir; çünkü zaten topların renkleri sabittir ve dahil olacakları sınıflar da bellidir.[13]

3.4.5. Ölçevler (Metrikler)

Gerek makine öğrenmesi gerekse derin öğrenme metotları olsun; eldeki verilerin sınıflandırılması istendiğinde, ilgili öğrenme metotları sonucunda veriler, çalıştırılan yapılarca belirli sınıflara atanmaya çalışılır. Bunun sonucunda doğru olup gerçekten de doğru sınıfa atan TP (True Positive – Doğru Pozitif), doğru zannedilip yanlış sınıfa atanan FP (False Positive – Yanlış Pozitif), aslında doğru olup doğru sınıfa

13 Araştırmacının kendi bilgi birikimine bağlı ifadelerdir.

atanamayan FN (False Negative – Yanlış Negatif); en sonunda da yanlış olduğu tespit edilen ve başarılı şekilde diğer sınıfa atanan TN (True Negative – Doğru Negatif) kavramları önümüze çıkmaktadır.[14]

Bu bahsettiğimiz TP, FP, FN ve TN değişkenleri; çeşitli formüllerde yer aldıklarında araştırmacıların yorum yapmalarını sağlayan metrikler, yani "ölçevler" sağlamaktadır. Bu çalışmada karşılaşılacak ölçütler şunlardır: Kesinlik (Precision), Hassasiyet (Recall), F1-Score, Destek (Support) ve Doğruluk (accuracy). Bunların arasında Accuracy, genel anlamda en çok dikkate alınan ölçüttür; çünkü kabaca şu demektir: "Çalıştırılan sistem, test edilen veriyi ne kadarlık bir oranda doğru tahmin edebilmiştir?"

İlgili ölçevler şu formüllerle hesaplanmaktadır:

$$F1 = 2 * \frac{Kesinlik * Hassasiyet}{Kesinlik + Hassasiyet}$$

Şekil 3.2 – F1 Score Formülü (Dinçer, Kayaoğlu ve Safarlı, 2022)

$$TP + FP = Toplam\ Pozitif\ Sınıflandırma$$

$$Kesinlik = \frac{TP}{Toplam\ Pozitif\ Sınıflandırma}$$

Şekil 3.3 – Kesinlik Formülü (Dinçer, Kayaoğlu ve Safarlı, 2022)

14 Wikipedia.org; "PRECISION AND RECALL"; https://en.wikipedia.org/wiki/Precision_and_recall, Erişim: 30.01.2024

$$Hassasiyet = \frac{TP}{TP + FN}$$

$$TP + FN = Toplam\ Asıl\ Sınıflandırma$$

$$Hassasiyet = \frac{TP}{Toplam\ Asıl\ Sınıflandırma}$$

Şekil 3.4 – Hassasiyet Formülü (Dinçer, Kayaoğlu ve Safarlı, 2022)

$$Doğruluk = \frac{TN+TP}{TP+FP+FN+TN}$$

Şekil 3.5 – Doğruluk Formülü (Göçgün ve Onan, 2021)

3.4.6. Makine Öğrenmesi

Yapay zekâ ve makine öğrenmesi, birbirlerine benzeyen ama farklı kavramlardır. Yapay zekâ, bilgisayar programların tıpkı insanların yaptığı gibi öğrenmesine ve buna göre işlem göstermesine; makine öğrenmesi ise bu amaca hizmet eden algoritmalara denmektedir. Makine öğrenmesi, ilgili amaca hizmet eden teknik ve algoritmaların üretilmesini amaçlayan bilimsel bir alandır. Makine öğrenmesi ile istatistik bilimine dayalı çalışmalar gerçekleştirilebilir. Tahminlemeler üzerine işlemler yapılabilir. Makineler, geçmişten öğrendikleri verilere göre oluşturulan bir model yardımıyla geleceği tahminlemeye çalışırlar. Makine öğrenmesi, istatistik gibi matematiksel disiplinlere dayalı geliştirilmiş bir metottur. Klasik yöntemlerde insan faktörünün devreye girmesiyle belirli kalıplarla işlem gerçekleştirilmesi süreci vardır; kaldı ki bu süreç, gerçekçilikten uzaktır. Makine öğrenmesinde ise makinenin kendisi sürecin üstünde hakimiyet kurmaktadır (Nacar ve Erdebilli, 2021).

3.4.7. TF-IDF (Terim Frekansı/Term Frequency & Ters Terim Frekansı/Inverse Document Frequency)

TF-IDF, sırasıyla Terim Frekansı (Term Frequency) ve Ters Terim Frekansı (Inverse Document Frequency) kavramlarını sembolize eder. TF-IDF ile bir dokümandaki kelimelerin incelenmesi söz konusudur. Terim Frekansı, bir dokümanda bir terimin kaç defa tekrar ettiği; Ters Terim Frekansı ise herhangi bir terimin birden fazla dokumanda yer alma sıklığı üzerine oluşan kavramlardır. Bir terim, bir dokümanda ne kadar sık tekrar ediyorsa o kadar değersiz; birden fazla dokümanda ne kadar çok yer alıyorsa o kadar değerlidir. Örneğin "ve" gibi bağlaçlar," ile" gibi edat ya da bağlaçlar vs. okuduğumuz herhangi bir metinde sıklıkla görebileceğimiz kelimelerdir. Ayrıca bu kelimeler, bir cümle içindeki duygu bağlamını etkilemezler. Bu tür sebeplerden dolayı; bilgi taşımayan, daha az değerli kelimeler olarak kabul edilirler (Hark ve Karcı, 2019).

3.4.8. Stop Words (Stopwords – Durak Kelimeler)

Stop Words kavramı da yukarıda açıkladığımız TF-IDF kavramından gelmektedir. Metin madenciliği çalışmalarında; özellikle duygu analizlerinde, sıklıkla tekrar eden ve analizi duygusal açıdan etkilemeyecek bu "durak kelimelerin", analizden çıkarılması gerektiği sonucuna varılmaktadır. İlgisiz olanı elemek, gerekli olana erişmek ve işlem yükünü hafifletmek amacıyla, doğal dil işleme uygulamalarında bu "durak kelimelerin" filtrelenmesi yaygın olarak gerçekleştirilen ilk adımlardan biridir (Kumova Metin ve Karaoğlan, 2017).

3.4.9. Tokenization (Tokenizasyon/Jetonlama)

Tokenizasyon terimi metin madenciliğinde kullanılan bir yöntemin adıdır. Veri güvenliği tehlikeye atılmadan, verinin bilgilerinin saklanarak özgün tanımlama sembolleriyle değiştirmesi demektir (Alagha, 2023). İki yöntemden bahsedilebilir; birincisi Kelime Tokenizasyonudur. Burada geniş hacimli ifadelerin, kelimelere bölünmesi söz konusudur. İkinci yöntem ise n-gram yöntemidir; burada ise kelimelerin birli, ikili, üçlü gruplandırılması akla gelmektedir.

3.4.10. Lemmatizasyon (Lemmatization – Kök Çözümleme)

Lemmatizasyon (Lemmatization) çekimli sözcüklerin "otomatik" bir şekilde, bir sözlükte geçtiği gibi basit hallerine dönüştürülmesi ya da indirgenmesi işlemine denir; aslında kök çözümleme işlemidir ve bir sınıflandırma türüdür. Bu basit biçime de Lemma adı verilir. Günümüzde metinlerin bilgisayar ortamında üretildiği ve bilgisayarların ne kadar hızlı çalışabildiği düşünülürse; insan eli yerine otomatik olarak bu işlemin gerçekleştirilmesi makul bir yöntemdir. Bu işlemle sözcük ailelerinin tespiti de mümkün hale gelmiştir (Tahiroğlu, 2021).

3.4.11. N-Gram kavramı

N-Gram Yönteminde N sayısı, bir metin içindeki öğelerin kaç tanesinin beraber bir bitişik dizisi oluşturduğuna karşılık gelen bir sayıdır (Alagha, 2023). CAVNAR ve TRENKLE, N-gram kavramını, "daha uzun bir dizenin N karakterli bir dilimi" olarak tanımlamışlardır. Bir dizide birlikte ortaya çıkan herhangi bir karakter kümesini tanımlamışlar ve bir dizinin üst üste binen N gramlara bölünebildiğini ifade etmişlerdir (Cavnar ve Trenkle, 1994).

CAVNAR ve TRENKLE, İngilizcedeki TEXT (Metin) kelimesini, ilgili çalışmalarında aşağıdaki şekilde n-gramlara bölmüşlerdir:

bi-gramlar: _T, TE, EX, XT, T_
tri-gramlar: _TE, TEX, EXT, XT_, T_ _
quad-gramlar: _TEX, TEXT, EXT_, XT_ _, T_ _ _
(Burada görülen altçizgiler (_) boşluğu temsil etmektedir.)

Anlaşılacağı üzere iki eleman bir araya geldiğinde, bi-gram, üç eleman bir araya geldiğinde tri-gram ve dört eleman bir araya geldiğinde de quad-gram oluşturmaktadır. Tek elemanlı n-gram ise bir uni-gram olarak adlandırılır.

Metin sınıflandırması çalışmalarında harf bazı yerine, kelime bazında düşünmek gerekmektedir. Çalışma sırasında metinlerin elemanları olan cümlelerin tek tek ögelere ayrılması sonucu her kelime, tıpkı TEXT örneğindeki gibi elemanlara dönüşür.

Eğer kelimeler tek tek alınırsa birer uni-gram, iki kelime bir araya alınırsa bi-gram, üç kelime bir arada alınırsa tri-gram ve dört kelime bir arada alınırsa, quad-gram elde edilmiş olunur (Cavnar ve Trenkle, 1994).

3.4.12. Derin Öğrenme

Derin öğrenme, bir yapay zekâ tekniğidir; hatta alt kümesidir denebilir, fakat farklı yeteneklere sahiptir. İnsan beyninin çalışma mantığını taklit eden bir işlev olarak yorumlanabilir (Taye, 2023). Makine öğrenmesinde bazı durumlarda dış müdahaleler, özellikle insan müdahalesi daha fazla gerekirken; derin öğrenme, kendi içindeki algoritmalarla öğrenerek hatayı düzeltebilmektedir. Bilgisayara aktarılan veri kümesi yardımıyla bilgilerin genelleştirilmesi esastır (Aalami, 2020). Derin öğrenme; metin madenciliği, spam mücadelesi, resim kategorizasyonu gibi çeşitli alanlarda kullanılmakta olan bir makine öğrenmesi yaklaşımıdır. Derin Öğrenme denince birden fazla katmandan oluşan öğrenme yapıları, dolayısıyla modelleri akla gelmektedir (Taye, 2023).

3.4.13. Yapay Sinir Ağları

Yapay sinir ağları, insan beynindeki sinirlerin yapısından esinlenilerek; matematiksel bir yapı kurulmasına dayanan bir yapay zekâ yöntemidir. Sadece şeklen değil, yetisel olarak da insan beynini; yani öğrenme ve hesaplama gibi yeteneklerini taklit eder, çıkarımlarda bulunur. Otomatik olarak, dış müdahaleye ihtiyaç duymadan bilgi türetme özelliğine sahip bir sistemdir. Örüntü (*Pattern*) tanıyabilir, sınıflandırma yapabilir, bilgi modelleyebilir, tahminleme ve optimizasyon gerçekleştirebilir; çözülmesi zor problemlerin çözümünde ciddi rol oynayabilirler. Varsayımsız, veri koruyan, deneyerek öğrenen modeller içermesi nedeniyle yüksek başarıya erişebilen bir araçtır (Çınaroğlu ve Avcı, 2020).

3.4.14. Epoch ve Kayıp (Hata) Fonksiyonu

Epoch, derin öğrenme modellerinde, eğitim verisinin yinelenmesi; başka bir tanımlamayla: veri kümesi bilgilerinin, kullanılan algoritma etrafındaki dönüşüne verilen addır. Bu geçiş, veri kümesinin algoritmanın başından sonuna, sonra tekrar

sondan başa dönüşünü ifade eder. Veri kümesindeki örneklemler, her epoch'ta modelin parametrelerini güncelleyebilir. Derin öğrenme algoritmaları, modeldeki hataları düzeltebilmek için işte bu geçişleri kullanmaktadır; belirli bir sayıda geçiş sağlandıkça hata düzeltmesi daha etkili olabilecektir. Bu süreçler, ayrıca grafiğe dökülebilmektedir.[15]

Epoch ile bağlantılı olan Kayıp ya da Hata Fonksiyonu (Loss) ise derin öğrenme modellerinin çıktı katmanlarında kullanılır. Tahmin edilen çıktıyı/etiketi ve beklenen çıktıyı/etiketi karşılaştıran model, elde edilen hata fonksiyonu sayesinde öngörülen ile gerçekte olan arasındaki farkı tespit edilebilmektedir. Model, bu farka göre geri bildirimde bulunacak, bir sonraki epoch için yeni parametrelerin belirlenmesi sağlanacaktır. Bu şekilde hata fonksiyonu değerinin belli bir epoch tekrarı sonrasında olabildiğince en az seviyeye inmesi beklenir (Taye, 2023).

3.4.15. K-Fold/N-Fold Cross Validation (N-Katlı Çapraz Doğrulama)

N-katlı (veya K-katlı) çapraz doğrulama, makine öğrenmesinde kullanılmakta olan bir yöntemdir. Veri kümesi n sayıda kata bölünür ve her yinelemede katlardan biri test, diğerleri ise eğitim kümesi olarak seçilir ve kullanılır. Her adımda bir sonraki kat test ve kalan diğer katlar, eğitim kümesine dönüşür; tüm veri kümesi bu şekilde incelenene kadar süreç bu şekilde devam eder. Amaç modelin doğruluğunun ve geçerliliğinin değerlendirilmesidir. Bu yöntem sayesinde modelin tahminlemesinin tamamen veri kümesine dayalı olup olmadığı test edilebilir. Ayrıca veri kümesinde homojen olmayan verilerin varlığına karşı da önlem alınmasını sağlayan bir yöntem kullanılmış olunur (Barut ve Altuntaş, 2023).

15 Simplilearn.com; "WHAT IS EPOCH IN MACHINE LEARNING?" https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning; Erişim: 31.01.2024

3.5. Veri kümesinin Ön İşlemesi

Web kazıma işlemlerinin tamamlanması, elde edilen verinin tek bir Excel dosyasında birleştirilmesi ve İspanyolca verilerin çevrilmesi gibi; yukarıdaki bölümde belirtilen işlemler sonrasında ilgili csv dosyasının Python ortamına alınarak esas çalışmanın başlamasına sıra gelmiştir.

Bu esnada veri kümesinde tam 50.000 satır ve 4 sütun bulunmaktadır. Bu 4 sütunda; her ürünün Amazon.com sitesindeki uzun adı, kullanıcı yorumunun başlığı, kullanıcıların verdiği puanlar ve kullanıcıların yaptığı yorumların metni, sırayla yer almaktadır. Kullanıcıların 1 ile 5 arasında puan vermiş oldukları bilgisi, önceki bölümlerde iletilmiştir. Tüm yorumlar ve puanlamalar anonim durumdadır, yani kimin hangi yorumu yaptığı belli değildir; ancak siteye girerek yapılacak bir araştırmayla tespit edilebilecek durumdadır. Ürünün adı ve yorum başlığı bilgilerini veri kümesinde, dolayısıyla bu çalışmada bulunması; nihai okuyucuyla paylaşılmayacak olsa da hem bu gizliliğin ihlali hem de duygu analizinde faydasız olmaları açısından gereksizdir. Bu nedenle ilgili Excel ortamında bu sütunlar elle silinmiştir. Kalan sütunların isimleri, daha rahat anlaşılabilmesi için; puanlamalar "label", yorumlar "text" olacak şekilde değiştirilmiştir. "Label" isminin verilmesinin sebebi, esasında yorumların olumlu mu yoksa olumsuz mu olduklarının bir etiketleme ile belirlenecek olmasıdır. Excel dosyasının csv'ye çevrilmesi sonrasında Python'ın net olarak algılayabilmesi için virgül gibi bir sembolle sütunlar arası verilerin ayrılması gerekir; bu nedenle yorum ve puanlardaki tüm virgül sembolleri silinmiş, sütunlar arasına virgüller getirilerek bu ayrım gerçekleştirilmiştir.

- Python'ın bir Excel dosyasını içeri alabilmesi için Pandas kütüphanesi ve şu komut satırı kullanılabilir:

data = pd.read_csv('Alınan_dosyanın_adresi\\dosyanın_adı.csv',delimiter=',', index_col=False)

Bunun öncesinde ilgili kütüphaneler Python'a çağrılmaktadır.

- Tüm çalışmalarda, yukarıda bahsedilen k-fold yöntemi kullanılmıştır. Bunun için şu komut satırı kullanılabilir:

kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=seed)

Buradaki 10 değeri, kat sayısının (K değeri) 10 olarak belirlendiğini göstermektedir.

- Aşağıdaki kod ile veri kümesinde ilgili puan değerlerine göre kaçar satır bulunduğu görülebilmektedir:

data['label'].value_counts()

Buna göre veri kümesinde 10.367 adet 1 puan, 7.375 adet 2 puan, 8.540 adet 3 puan, 10.151 adet 4 puan ve 13.567 adet 5 puan içeren satır yer almaktadır.

- Analizin iki kutuplu yapılacağı; dolayısıyla 1'den 5'e kadar olan puanlamaların analiz açısından verimli olmayacağı düşünülerek aşağıdaki kodlarla 1 ve 2'ler 0 (olumsuz), 4 ve 5'ler 1 (olumlu) değerlerine döndürülmüş; 3 değerlerini içeren tüm satırlar veri kümesinden çıkarılmıştır.

data['label'].replace(1,value=0,inplace=True)

data['label'].replace(2,value=0,inplace=True)

data['label'].replace(4,value=1,inplace=True)

data['label'].replace(5,value=1,inplace=True)

data = data[~data['label'].isin([3])]

Bu işlem sonrasında 0 değerine sahip 17.742; 1 değerine sahip 23.718; toplamda da 41.460 satır kaldığı, aynı "value.counts" koduyla tespit edilmiştir.

- Sonrasında, aşağıdaki kod silsilesiyle boşluklar, *NaN* ("not a number" – "bir sayı değildir") tipi değerlerle doldurulmuş, bu *NaN* içeren satırlar veri kümesinden atılmış; kelimeler küçük harflere dönüştürülerek ayrılmış, alfabetik olmayan, emoji gibi analize katkıda bulunmayacak semboller veri kümesinden atılmış, yukarıdaki bir bölümde de bahsedilen "Stop Words" tipi kelimeler de veri kümesinden çıkarılmış ve kelimeler yeniden kümede birleştirilmiştir. Sonrasında boşlukların *NaN* ile doldurup tüm *NaN*'ların atılması işlemi tekrarlanmıştır.

data["text"] = data["text"].fillna("")

data["text"].replace('', np.nan, inplace=True)

data.dropna(subset=["text"], inplace=True)

data["text"] = data["text"].apply(lambda x: " ".join(kelime.lower() for kelime in x.split()))

data['text'] = data['text'].str.replace('[^\w\s]','')

data['text'] = data['text'].str.replace('\d','')

data['text'] = data['text'].apply(lambda x: " ".join(kelime for kelime in x.split() if kelime not in stopwords))

data['text'] = data['text'].apply(lambda x: " ".join((kelime) for kelime in x.split()))

Bu işlem sonrasında 0 değerine sahip 17.628; 1 değerine sahip 22.957; toplamda da 40.585 satır kaldığı, yine value.counts koduyla tespit edilmiştir.

- Bundan sonra ise sıra, yukarıdaki bölümlerde açıklanan TF-IDF işleminin yapılmasına gelmiştir.

cv = TfidfVectorizer(max_features = 1000, ngram_range=(1,1))

Çalışmadaki tüm algoritmalarda, öznitelik olarak alınan terim sayısı 1000 olarak belirlenmiştir. Tüm makine öğrenmesi algoritmalarında; önceki bölümlerde açıklanan n-gram metodu kullanılmıştır. Buna göre "ngram_range" ifadesiyle kullanılan: (1,1) sadece uni-gram; (1,2) uni-gram ve bi-gram; (1,3) uni-gram, bi-gram, tri-gram; (2,2) sadece bi-gram; (2,3) bi-gram ve tri-gram; (3,3) ise sadece tri-gram kullanıldığını göstermektedir.

Ön İşleme adımları bu şekilde sona ermektedir. Derin öğrenmede ise n-gram adımları yoktur.

3.6. Çalışmada Kullanılan Algoritmalar

3.6.1. Makine Öğrenmesi Algoritmaları

K-EN YAKIN KOMŞU (KNN – KEYK) ALGORİTMASI

KNN (K-Nearest Neighbor / K-En Yakın Komşu – KEYK olarak da adlandırılmaktadır), bir denetimli öğrenme algoritmasıdır ve sınıflandırma için kullanılır. Sınıflandırma sonucu elde edilen niteliklere uygun olarak; sınıflandırılmak istenen yeni bireyin önceki tüm bireylere olan uzaklığı dikkate alınarak en yakın k sınıfı kullanılır. Bu işlem sonucunda test verilerinin aitliği, en yakın değer hangi sınıfa mensup ise bu sınıfın doğru sınıf olabileceği görüşüne dayalıdır. Mesafe hesaplanırken hangi algoritmanın kullanılacağı ve komşu sayısının kaç olacağı, belirlenmesi gereken önemli optimizasyon parametreleridir. En uygun k sayısı, ilgili araştırmacının yapacağı deneylerle belirlenebilecektir (Sabancı, 2016).

(K değeri, kullanıcının yapmak istediği işleme göre değişebilmektedir; yapılan çalışmalarda genel olarak k=3 değerinin yaygınlıkla kullanıldığı gözlemlenmektedir.)[16]

Mesafe hesaplamasında Öklid Teoremi'ne dayalı bir yöntem kullanılmaktadır.

Öklid hesaplama yöntemi:

$$d(x_i, x_j) = \left(\sum_{s=1}^{p} (x_{is} + x_{js})^2 \right)^2$$

xi ve xj aralarındaki mesafe öğrenilmek istenen iki noktadır.

Şekil 3.6 – Öklid Formülü (Sabancı, 2016).

16 Araştırmacının kişisel gözlemlerine dayanmaktadır.

LOJİSTİK REGRESYON ALGORİTMASI

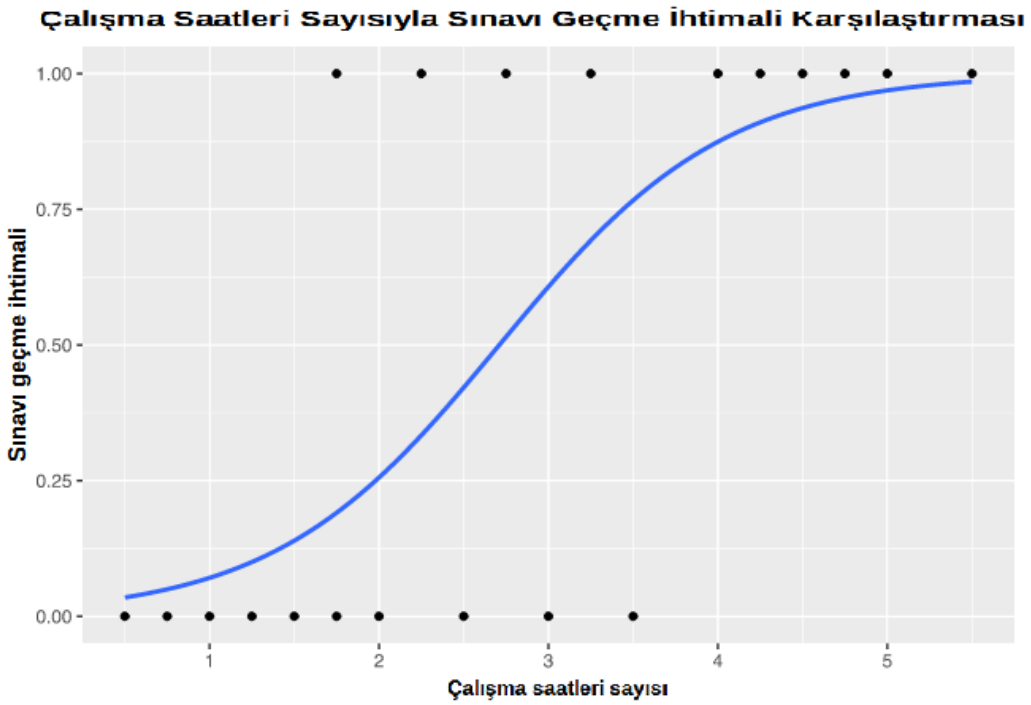
Bir araştırma konusu ile ortaya çıkan değişkenler arası ilişkilerin en iyi şekilde tanımlanabilmesi için hangi yöntemin kullanılacağı; söz konusu değişkenlerin sürekli ya da süreksiz olmalarına göre değişebilir. Regresyon analizi, değişkenler arasındaki söz konusu ilişkiler incelenirken kullanılan istatistiksel yöntemlerden biridir. Temel amaç; yordanan ve yordayıcı değişkenler arası ilişkinin, en az değişkenin en uyumlu şekilde tanımlandığı bir model kurmaktır. *(Yordamak; "bilinen veya gözlenen durumlardan yola çıkarak bilinmeyen veya gözlenmeyen durumlar hakkında tahminde bulunmak" anlamına gelmektedir.)*[17] Öngörülmek istenen durum, süreksiz bir durumdur. Lineer regresyon, olasılığa dayalıdır. Olasılık; basitçe belirli sonuç sayısının toplam olası sonuçlar içerisindeki oranına denmektedir. Örneğin, bir zar bir kez atıldığında herhangi bir sayı gelme olasılığı 1/6'dır; çünkü zarda altı sayı ve altı olası sonuç vardır (Şenel ve Alatlı, 2014).

İstatistikte lojistik model (veya logit model), bir olayın log-olasılıklarını (log-odds) bir veya daha fazla bağımsız değişkenin doğrusal kombinasyonu olarak modelleyen istatistiksel bir modeldir. Logit, p'nin bir olasılık olduğu p/ (1 − p) oranlarının logaritmasına eşit olduğu durum için kullanılan bir kavramdır ve log-odds olarak da adlandırılır. Regresyon analizinde **lojistik regresyon** (veya logit regresyon), bir lojistik modelin parametrelerinin (doğrusal kombinasyondaki katsayılar) tahmin edilmesidir. Log-odds ölçeğinin ölçüm birimine logit adı verilir, bu ad İngilizcedeki **Log**istics Un**it** (Lojistik Birim) kelime öbeğinden gelmektedir.[18]

17 Sozluk.gov.tr; "YORDAMAK"; https://www.sozluk.gov.tr/; Erişim: 29.01.2024

18 Wikipedia.org; "LOGISTIC REGRESSION"; https://en.wikipedia.org/wiki/Logistic_regression; Erişim 29.01.2024

Aşağıdaki şekilde bir lojistik eğrisi örneği görülmektedir. Dikey eksende sınavı geçme olasılığı, yatay eksende de sınava girecek öğrencinin çalıştığı saat sayısı görülmektedir. Öğrencinin ilk saatlerindeki verimlilik, orta zaman diliminden en yüksek düzeye ulaşmakta, son saatlerinde de yeniden düşmektedir. Nihayetinde en yüksek geçme olasılığına ulaşılmaktadır.[19]



Şekil 3.7 – Lojistik Regresyon Formülü (Wikipedia)

NAIVE BAYES ALGORİTMASI

Naive Bayes sınıflandırma algoritması, makine öğrenmesi çalışmalarında sıklıkla kullanılan; kolay ve anlaşılır bir algoritmadır. Basit verilerle hızlı ve yüksek doğrulukta sonuçlar üretebilmektedir (Şahinaslan ve Dalyan, 2022). Basit yapılı bir istatistiksel tahmin algoritmasıdır. Prensip olarak her özelliğin bağımlılığını yalnızca bir sınıfa kaydederek optimum sınıflandırmayı gerçekleştirir. İlk olarak Thomas Bayes tarafından önerilen Bayes kuralı, aynı stokastik sürece bağlı iki farklı ardışık olayın (X ve Y) marjinal ve koşullu olasılıkları arasındaki ilişkiyi açıklar (Orhan, Adem ve Cömert, 2012).

19 Wikipedia.org; “LOGISTIC REGRESSION”; https://en.wikipedia.org/wiki/Logistic_regression; Erişim 29.01.2024

(Stokastik veya rastgele bir süreç; olasılık teorisi ve ilgili alanlarda, genellikle bir olasılık uzayındaki rastgele değişkenlerin dizisi olarak tanımlanan matematiksel bir nesnedir).[20]

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

Şekil 3.8 – Naive Bayes Formülü (Orhan, Adem ve Cömert, 2012).

Yukarıdaki şekilde görülen P(X) ve P(Y) marjinal olasılıklardır; burada P(X|Y) ve P(Y|X) koşullu olasılıklardır. Sınıflandırmada Bayes kuralı kullanıldığında, çıktı için karar sınıfı olarak tüm olası çıktı durumları arasında maksimum olasılığa sahip olan durum seçilir (Orhan, Adem ve Cömert, 2012).

RASTGELE ORMANLAR (RANDOM FORESTS) ALGORİTMASI

2001 yılında Leo Breiman tarafından geliştirilen Rastgele Ormanlar yöntemi, rastgele seçilen verilerin alt uzaylarında büyüyen karar ağaçlarıyla tahmin kümesi oluşturmaya dayalı bir makine öğrenme metodudur. Oldukça başarılı ve hızlı sonuçlar verebilen bir metot olmasından dolayı sık tercih edilir. Hem kategorik hem sürekli verilerin bulunduğu; aynı zamanda farklı boyutlardaki veri kümelerinde karar ağacı olarak kullanılabilmektedir. Neden sonuç ilişkisini belirlemede etkili bir tahminleyicidir. Yapı içinde yer alan her düğümden rastgele seçilen en iyi değişken kullanılır; bu yapılırken CART (Classification and Regression Tree / Regresyon ve Sınıflandırma Ağacı) adı verilen bir karar ağacı yapısı kullanılmaktadır. CART analizinde düğümdeki dallar GINI katsayısına göre ikili gruplara ayrılır. *(GINI katsayısı,*

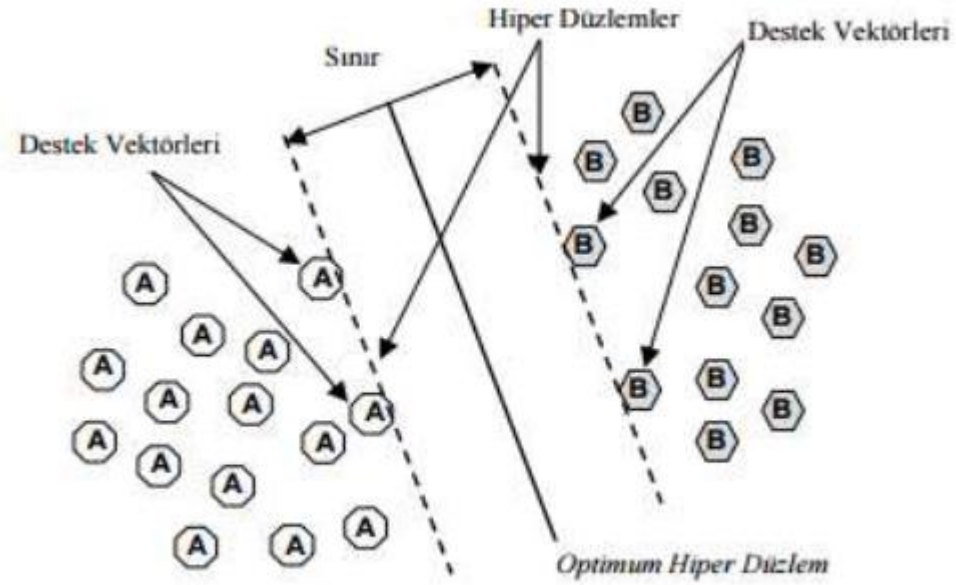
20 Wikipedia.org; “STOCHASTIC PROCESS”; https://en.wikipedia.org/wiki/Stochastic_process; Erişim: 29.01.2024

sınıfların homojenliği hakkında bilgi veren bir ölçüdür. Katsayının küçüklüğü kategorinin homojen olduğunu göstermektedir).

Rastgele Orman Modeli; veri kümesinin tamamı ya da eğitim ve test olarak bölünmüş farklı gruplarla kurulabilir. Eğitim kümesinde bulunan tüm değişkenler arasından en iyi bölünmeyi sağlayacak olanları belirlemek için belli sayıda rastgele örnek seçilir; sonra yine belli bir sayıda karar ağacının tahminleri toplanır. Oy çoğunluğu parametresi göz önüne alınır ve yeni bir kümesi tahmininde bulunulur. Oy çoğunluğuna göre verinin sınıfı belirlenmiş olur (Parlak ve Kayri, 2022).

DESTEK VEKTÖR MAKİNELERİ (DVM – SUPPORT VECTOR MACHINES – SVM)

Destek Vektör Makineleri'nin (DVM – Support Vector Machines – SVM) temelini, Corinna CORTES ve Vladimir VAPNIK'in çalışmaları oluşturmaktadır; kendileri 1995 yılında "Support-Vector Networks" (Destek Vektör Ağları) adlı bir makale yayımlamışlardır. Modelleri, düşey algoritma çalışmaları yaparak Hyperplane (hiper düzlem) adlı kavrama dayalı doğrusal sınıflandırma yapabilmektedir. CORTES ve VAPNIK regresyon problemleri için bir başka algoritma daha geliştirmişlerdir. SVM, özellikle örüntü (pattern) tanınmasında sıklıkla kullanılmaktadır. Hedef, sınıfları ayırabilen en uygun Hyperplane'i bulabilmektir. Sınıflar arası sınırın azami hiper düzlemi, ideal hiper düzlem olarak söylenebilir (Cortes & Vapnik, 1995).



Şekil 3.9 – Destek Vektör Makineleri Tasarımı (Atasoy ve Tabak, 2018).

Şekilde görüldüğü üzere, iki farklı gruba dayalı öğeler bulunmaktadır. Grupların diğer gruba en yakın elemanları arasındaki eşit mesafenin tam ortasından geçen bir çizgi oluşmaktadır. İşte Hiper Düzlem adı, bu çizgiye verilmektedir. Bu düzlem, öğelerin net bir şekilde sınıflandırılmasını sağlamaktadır.[21]

3.6.2. Derin Öğrenme Algoritmaları

EVRİŞİMSEL SİNİR AĞI – (CNN – CONVOLUTIONAL NEURAL NETWORK)

CNN algoritması, bir yapay sinir ağı türüdür. CNN'de, genel sinir ağı yapısı mantığından farklı olarak Evrişim (Convolution) denilen bir yöntem kullanılmaktadır. Matematikte evrişim kavramı; örneğin f ve g olarak iki farklı fonksiyon alındığında; bu iki fonksiyonun, üçüncü bir fonksiyon üretmesinden ileri gelir. Hem elde edilen

21 Araştırmacının kendi derlemesidir.

sonuç fonksiyonunun kendisi hem de bu fonksiyonun nasıl hesaplandığı, evrişimin konularıdır.[22]

Mohammad TAYE, 2023 yılında yayımlanan "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions" adlı makalesinde CNN tanımını şöyle yapmıştır:

"Evrişimli sinir ağları (CNN'ler), nesneleri tanımlayabilen, tanıyabilen ve sınıflandırabilen, ayrıca görüntülerdeki nesneleri algılayıp bölümlere ayırabilen çok katmanlı sinir ağlarına dayanan yapay zekâ sistemleridir."

CNN, özellikle çeşitli 2 boyutlu nesneler üzerine çalışabilmesi için tasarlandığından; görsel ayırt etme, doğrusal programlama, metin sınıflandırması gibi pek çok uygulamada kullanılabilmektedir.

CNN Katmanları şunlardır: Evrişimli Katman (Convolutional Layer), Havuzlama Katmanı (Pooling Layer), Aktivasyon Fonksiyonu (Activation Function) ve Tamamen Bağlantılı Katman (Fully-Connected Layer). Bunlar öncesinde bir giriş ve sonrasında da bir çıkış katmanı yer almaktadır. Giriş katmanı, verilerin içeriye ilk alındığı; çıkış katmanı ise sonucun alındığı katmanlardır. Giriş ve Çıkış katmanı arasındaki katmanlara "Gizli Katmanlar" adı da verilmektedir (Taye, 2023).

Evrişim katmanı; ilk aşamada uygulanan bir dizi filtre veya çekirdekten oluşur. Her çekirdeğin belirli boyutları ve ağırlık değerleri vardır; bunlar sayesinde giriş katmanından öznitelikler ortaya konur. Veriler bir sonraki katmana aktarıldıkça ve geriye doğru geri bildirimler alındıkça bu ağırlık değerleri güncellenir (Taye, 2023).

Havuzlama katmanı, down-sampling, yani aşağı yönlü örnekleme yapmaktadır; yani gelen verilerin özelliklerini küçültmektedir. Bunun yapılabilmesi için kullanılan filtre; giriş verilerine asgari değer, azami değer ve ortalama değer yöntemleriyle havuzlama işlemlerini uygulamaktadır. Özellikle görüntü işlemede yaygın kullanıldığı

22 Wikipedia.org; "CONVOLUTION"; https://en.wikipedia.org/wiki/Convolution; Erişim: 30.01.2024

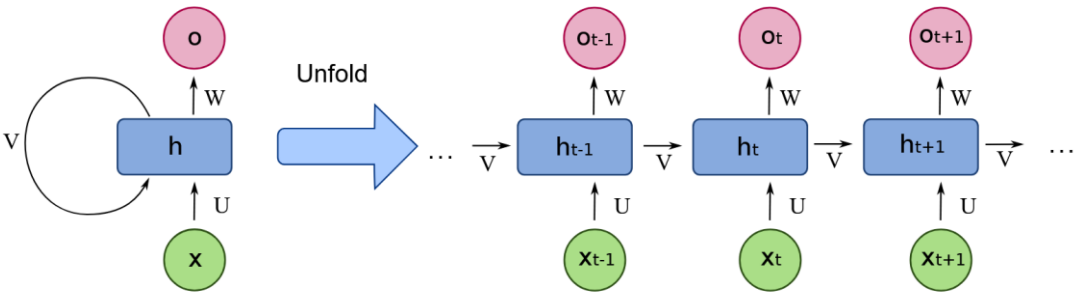
görülebilecek bir işlem olan havuzlamada, en çok azami (maksimum) değer yöntemi kullanıldığı gözlemlenebilir (Taye, 2023).

Tam Bağlantılı katmanlarda ise nöron adı verilen, gruplar halindeki yapılar bulunmaktadır; bunlar düğümler halinde birbirlerine bağlanırlar. Havuzlama katmanıyla da bağlı olan bu yapıların, çok detaylı hesaplamalarla, ihtiyaç duydukları eğitime erişimleri sağlanmaktadır (Taye, 2023).

RECURRENT NEURAL NETWORK (RNN – TEKRARLAYAN SİNİR AĞI)

RNN; zamanla genişleyen, sonraki adım zamanını besleyen kenarlara sahip bir sinir ağı türüdür. Bir metni ya da konuşma sinyalini, aslında bir dizi halindeki veriyi tanıyabilen bir yapıdır. Kısa yapıdaki hafızayı destekleyen döngüler, bu ağ içinde yer almaktadır. Giriş dizisi için belirli bir zaman tanımlanmadığından; girişin bir ağaç biçiminde hiyerarşik olarak işlenmesinin gerektiği yapıdır. RNN'de giriş verileri, hiyerarşik bir oluşum içinde bir ağaç yapısı gibi dallanarak işleme alınır (Abiodun ve diğerleri, 2018). Aslında RNN, zaman serisi verilerini kullanır. Tıpkı CNN'de olduğu gibi; öğrenebilmek için eğitim verisine ihtiyaç duymaktadır. Hafızası, ayırt edici özelliğidir. Genelde sinir ağlarında, girdi ve çıktı bağımsızlığı kavramı baskın iken; RNN tipi ağlarda çıktı, önceki öğelere bağlıdır. Tıpkı bir deyimin anlaşılabilmesi için cümle öğelerinin bilinen sırayla söylenmesi gerektiği gibi, RNN'de de öğelerin sırası önem taşımaktadır; çünkü bir önceki öğeye göre sonraki öğeyi tahminlemeye çalışan bir yapıdır. RNN'nin bir başka karakteristik özelliği ise; tüm katmanlarında aynı ağırlık parametresini paylaşmasıdır.[23]

23 Ibm.com; "WHAT ARE RECURRENT NEURAL NETWORKS?" https://www.ibm.com/topics/recurrent-neural-networks; Erişim: 30.01.2024



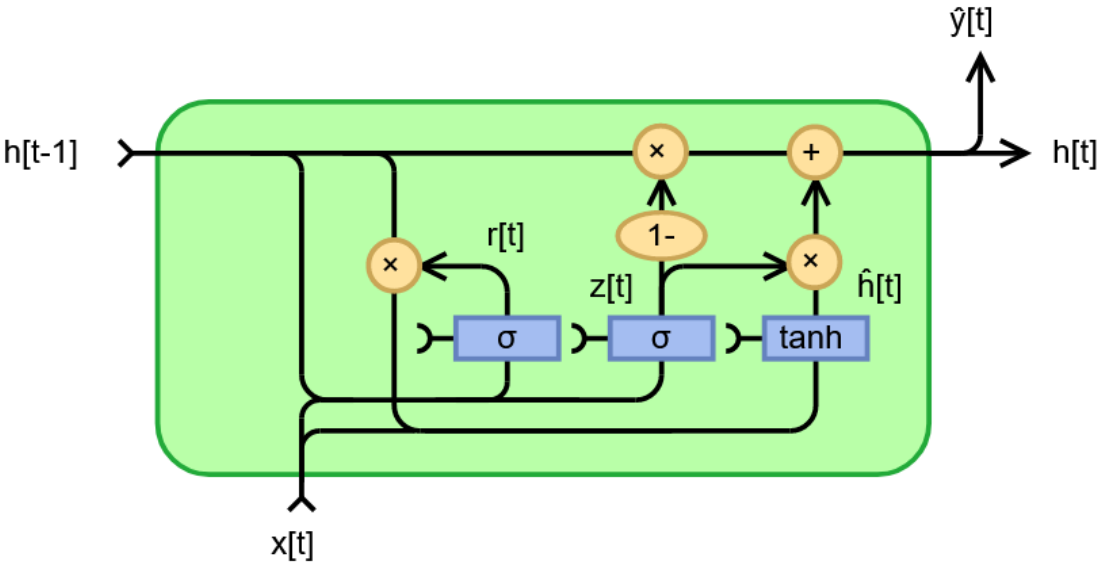
Şekil 3.10 – Recurrent Neural Network Modeli[24]

GATED RECURRENT UNITS (GRU – GEÇİTLİ TEKRARLAYAN BİRİMLER)

Geçitli Tekrarlayan Birimler (Gated Recurrent Units – GRU), 2014 yılında Kyunghyun Cho ve diğerleri tarafından ortaya konmuş bir RNN tipi bir mimari yapısıdır. LSTM mimarisine benzer bir yapısı vardır ama LSTM'den daha az parametreye sahiptir. Polifonik müzik, konuşma sinyali modellemeleri ve doğal dil işleme gibi işlemleri gerçekleştirebilir[25]. Modelde, birden çok zaman dahilindeki bağlamı hatırlayabilen, bilgi transferini sağlayan bir geçit yapısı bulunur. Geçmişten gelen hangi bilgilerin saklanıp hangilerinin unutulacağını belirleyen birer güncelleme ve sıfırlama kapısı vardır. LSTM'den daha az parametreye sahip olması ve LSTM'de de bulunan bu güncelleme ve sıfırlama geçitlerini tek bir yapı altında birleştirebilmesi sebebiyle; LSTM mimarisine göre daha iyi sonuçlar aldığı gözlemlenmiştir (Yurtsever, 2021).

24 Wikimedia.org; "RECURRENT NEURAL NETWORK UNFOLD"; https://upload.wikimedia.org/wikipedia/commons/b/b5/Recurrent_neural_network_unfold.svg; Erişim: 30.01.2024

25 Wikipedia.org; "GATED RECURRENT UNIT"; https://en.wikipedia.org/wiki/Gated_recurrent_unit; Erişim: 30.01.2024

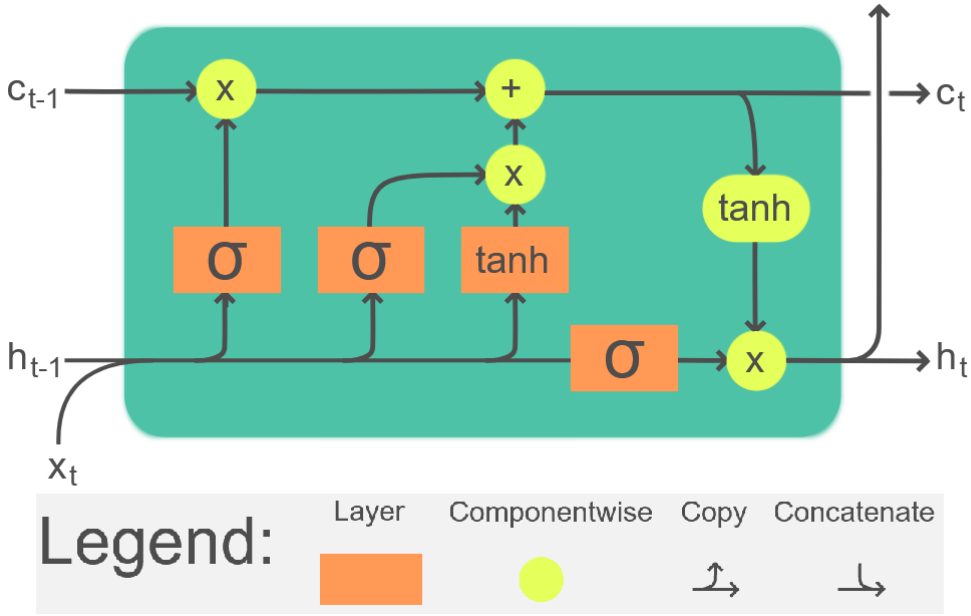


Şekil 3.11 – Gated Recurrent Unit (GRU) Modeli [26]

LONG SHORT-TERM MEMORY (LSTM – UZUN KISA-SÜRELİ HAFIZA)

LSTM, RNN mimarisi daha iyi hale getirmek için 1997 yılında Hochreiter ve Schmidhuber tanıtılmış bir RNN türüdür. RNN'den farklı olarak, kaybolan gradyanlar problemine çözüm getirebilmesi için belli bilgilerin ne zaman kullanılıp kullanılmayacağını belirleyebilecek şekilde geliştirilmiştir. LSTM de tıpkı bir CNN gibi giriş katmanı, gizli katmanlar ve çıkış katmanları içerir. Çeşitli bloklardan oluşan yapıda, bellek hücreleri, çarpan birimleri, giriş, çıkış ve unutma geçitleri vardır. Bahsedilen bu bileşenler ile bir önceki hücreden sonrakine ne kadar bilginin unutulacağına ne kadarının ise aktarılması gerektiğine karar verilir. Geriye dönük amaçla, gelecek verileri kullanamayan LSTM mimarisi için Schuster ve Paliwal, 1997 yılında çift yönlü bi-LSTM mimarisini ortaya koymuşlar; böylece iki yönlü bir trafik oluşturarak gelecekten gelen bilgilerle geçmişi güncelleyebilmeyi amaçlamışlardır (Yurtsever, 2021).

26 Wikipedia.org; "GATED RECURRENT UNIT"; https://en.wikipedia.org/wiki/Gated_recurrent_unit; Erişim: 30.01.2024



Şekil 3.12 – Long Short–Term Memory Modeli [27]

3.7. Algoritmaların Uygulanması ve Ulaşılan Değerler

3.7.1. Algoritmaların Uygulanması

Önce makine öğrenmesi algoritmaları, yani K-En Yakın Komşu (KNN), Lojistik Regresyon (LR), Naive Bayes (NB), Rastgele Ormanlar (RF) ve Destek Vektör Makineleri (SVM); sonrasında da derin öğrenme algoritmaları, yani Evrişimli Sinir Ağları (CNN), Geçitli Tekrarlayan Birimler (GRU), Uzun Kısa-Süreli Hafıza (LSTM) ve Basit Tekrarlayan Sinir Ağı (Simple RNN – RNN) uygulanmıştır.

27 Wikimedia.org.; "LSTM CELL"; https://upload.wikimedia.org/wikipedia/commons/9/93/LSTM_Cell.svg; Erişim: 30.01.2024

%60 doğruluk elde edilen, 1000 öznitelik ve uni-gram kullanılan KNN algoritmasının modelleme kodları aşağıdaki gibidir:

```
knn = KNeighborsClassifier()
for train, test in kfold.split(data.text, y):
    X_train = cv.fit_transform(data.text[train]).toarray() # bağımsız değişkenler
    X_test = cv.transform(data.text[test]).toarray() # bağımsız değişkenler
    knn.fit(X_train, y=y[train])
    predicted = knn.predict(X_test)
    accuracyScores.append(accuracy_score( y[test], predicted))

print(f"Average accuracy is: {np.mean(accuracyScores):.2f}")
```

```
Average accuracy is: 0.60
```

Şekil 3.13 – KNN Algoritması Modelleme Kodları Python Görüntüsü

Bu ilgili kodlar içinde yer alan komutlardan "fit", modelin eğitilmesi; "transform", verilerin sayısal değerlere çevrilmesi; "predict", ilgili algoritmanın tahminlemede bulunması; "append" ise daha önceden tanımlanan boş bir diziye doğruluk değerlerinin yerleştirilmesi için kullanılır. Bu değerlerin ortalaması alınarak ulaşılan ortalama doğruluk değeri, kullanılan makine öğrenmesi modelinin esas doğruluk değeri olarak kabul edilir.

Makine öğrenmesi algoritmalarında TF-IDF kullanılırken, derin öğrenme algoritmalarında ise Tokenizer (önceki bölümlerde anlatılan Tokenizasyon) metodu kullanılmıştır.

```
MAX_LENGTH = 200
tokenizer = Tokenizer()
tokenizer.fit_on_texts(data.text)
post_seq = tokenizer.texts_to_sequences(data.text)
post_seq_padded = pad_sequences(post_seq, maxlen=MAX_LENGTH)
post_seq_padded = np.array(post_seq_padded)

y = np.asarray(y)
vocab_size = len(tokenizer.word_index) + 1
```

Şekil 3.14 – Tokenizer Kodları Python Görüntüsü

LSTM derin öğrenme algoritmasının ilgili modelinin kod görüntüsü ise aşağıdaki gibidir:

```
histories = []
for train, test in kfold.split(post_seq_padded, y):
    model=Sequential()
    model.add(Embedding(vocab_size,200,input_length=MAX_LENGTH))
    model.add(LSTM(100,dropout=0.3))
    model.add(Dense(50,activation='relu'))
    model.add(Dropout(rate = 0.3))
    model.add(Dense(2, activation='sigmoid'))
    model.compile(optimizer=opt,loss='binary_crossentropy',metrics=['accuracy'])

    # Fit the model with early stopping
    history = model.fit(post_seq_padded[train], y=to_categorical(y[train]), batch_size=16,
    verbose=2, epochs=10, validation_data=(post_seq_padded[test], to_categorical(y[test])))
    histories.append(history)

    predicted = model.predict(post_seq_padded[test])
    predicted = np.argmax(predicted, axis=1)
    accuracyScores.append(accuracy_score( y[test], predicted))
    precisionScores.append(precision_score( y[test], predicted))
    recallScores.append(recall_score( y[test], predicted))

model.summary()
```

Şekil 3.15 – LSTM Algoritması Modelleme Kodları Python Görüntüsü

Burada önce model kurulmakta, önceki bölümlerde bahsedilen gizli katmanlar oluşturulduktan sonra ise çıkış katmanını simgeleyen "Dense" katmanı tanımlanmaktadır. Aradaki "Dropout" parametresi, modelin aşırı öğrenmesini engelleyen bir yöntemi temsil eder. Model tanımlandıktan sonra yine "fit" komutuyla eğitim gerçekleşmekte; yalnız burada makine öğrenmesinden farklı olarak, önceki bölümlerde bahsedilen "Epoch" sayısı da tanımlanmaktadır. Bazı derin öğrenme algoritmalarında, harcanan zaman bakımından 5, bazılarında ise 10 epoch uygulanmıştır. Kodun görülen son paragrafında ise; tıpkı bir önceki makine öğrenmesi kodu örneğinde olduğu gibi, tahminleme yapan ve bulunan doğruluk değerlerini ortalaması alınabilmesi için boş listeye doldurulan komutlar bulunmaktadır.

Modellerde görülen "x" değerleri ilgili verileri; "y" değerleri ise bunların sınıflandırılacağı etiketleri temsil etmektedir.

Derin öğrenmede yer aldığı daha öncede belirtilen Epoch'ların süreçleri, Python ortamında aşağıdaki şekilde gözlemlenebilmektedir:

```
Epoch 1/5
519/519 - 119s - loss: 0.4676 - accuracy: 0.7704
Epoch 2/5
519/519 - 120s - loss: 0.3266 - accuracy: 0.8649
Epoch 3/5
519/519 - 120s - loss: 0.2901 - accuracy: 0.8817
Epoch 4/5
519/519 - 120s - loss: 0.2544 - accuracy: 0.9006
Epoch 5/5
519/519 - 128s - loss: 0.2174 - accuracy: 0.9169
260/260 [==============================] - 10s 38
```

Şekil 3.16 – Epoch Süreçleri Python Görüntüsü

Bu şekilde her epoch'ta ulaşılan Loss (kayıp fonksiyonu) ve Accuracy (doğruluk) değerleri gözlemlenebilmektedir. Bu değerler, başka kodlarla saklanarak ileride bahsedeceğimiz grafiklere veri sağlamıştır. Kaç "fold" (kat) ile çapraz değerleme yapılıyor ise belirlenen epoch sayısı da o kadar tekrar etmektedir (Örneğin: 5 fold ve 5 epoch belirlenirse; toplamda 25 kez işlem tekrar eder).

Python'ın Scikit kütüphanesindeki "metrics" modülünün, önceki bölümlerde tanımlanan ölçevleri gösteren ilgili komutu her algoritma için kullanılmıştır. KNN için kullanılan kod, aşağıdaki şekilde örneklenmiştir; burada ilgili ölçevlerin ulaşılan değerleri görülebilmektedir:

```
1 print(metrics.classification_report(y[test], predicted))
```

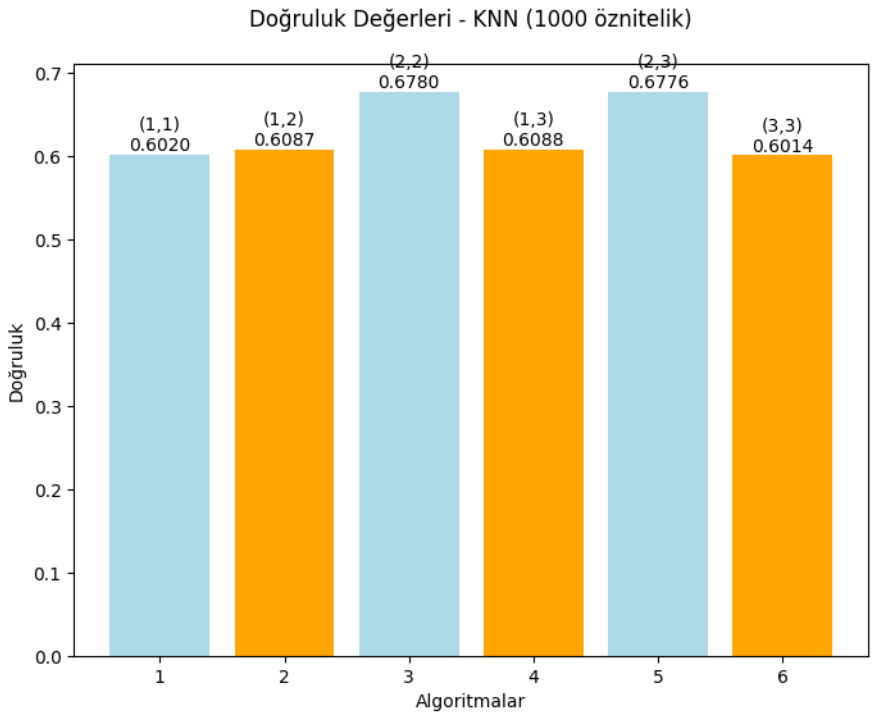
```
              precision    recall  f1-score   support

           0       0.72      0.14      0.24      1762
           1       0.59      0.96      0.73      2296

    accuracy                           0.60      4058
   macro avg       0.66      0.55      0.48      4058
weighted avg       0.65      0.60      0.52      4058
```

Şekil 3.17 – Ölçev Sınıflandırma Raporu Python Görüntüsü

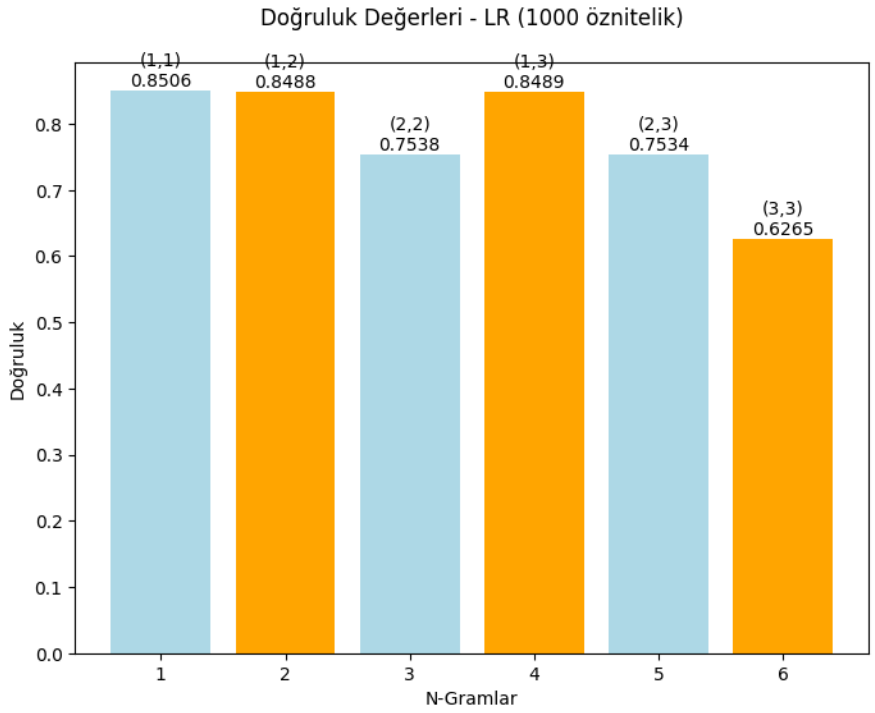
3.7.2. Ulaşılan Değerler

Bahsedildiği üzere tüm algoritmalar sonucunda çeşitli ölçevlere ulaşılmıştır; bu ölçevler arasında en kritik olanının, Doğruluk (Accuracy) değeri olduğu söylenebilir.

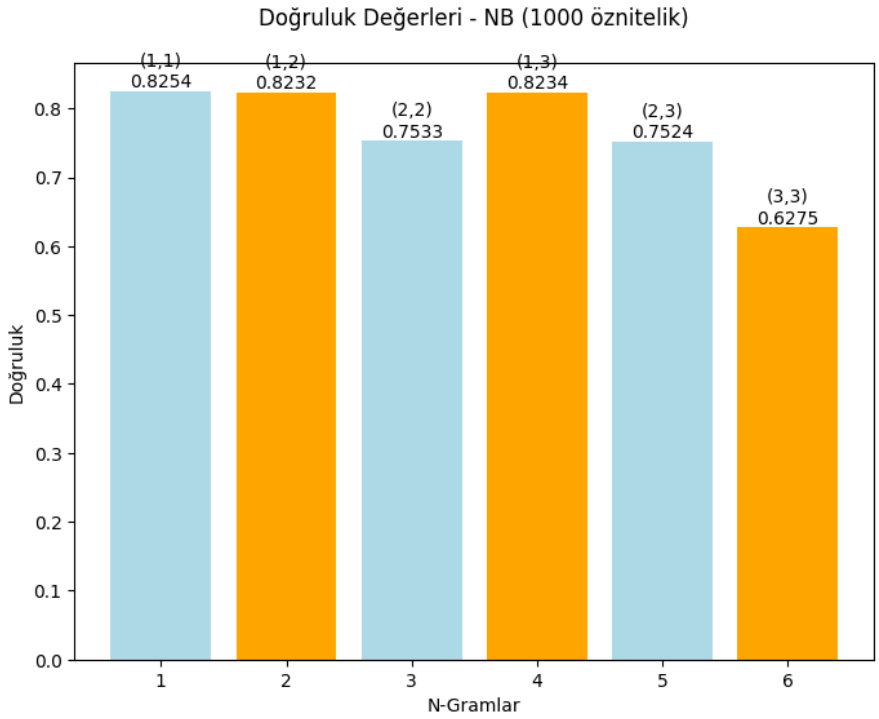
Tüm modellerin ulaştığı doğruluk değerleri aşağıdaki görsellerde yer almaktadır:



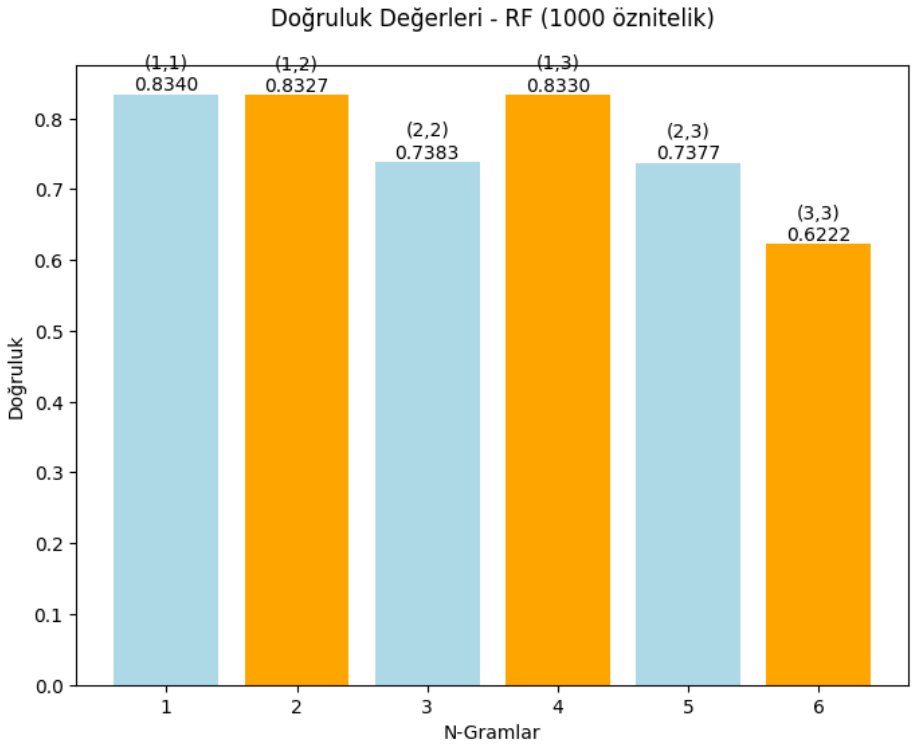
Şekil 3.18 – K En Yakın Komşu Doğruluk Değerleri Tablosu



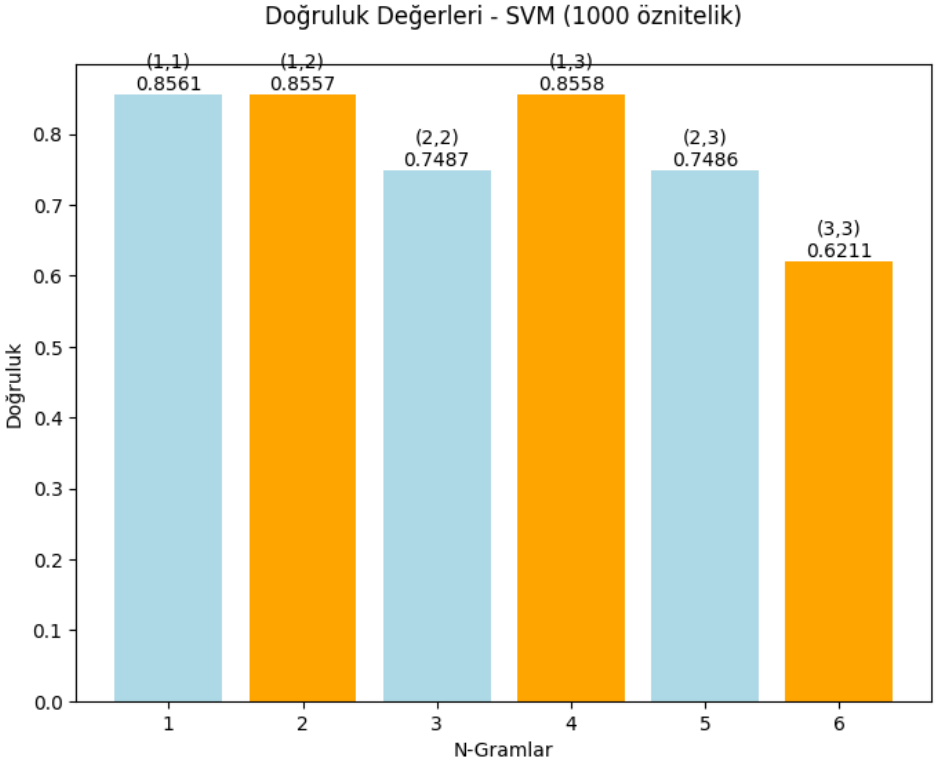
Şekil 3.19 – Lojistik Regresyon Doğruluk Değerleri Tablosu



Şekil 3.20 – Naive Bayes Doğruluk Değerleri Tablosu

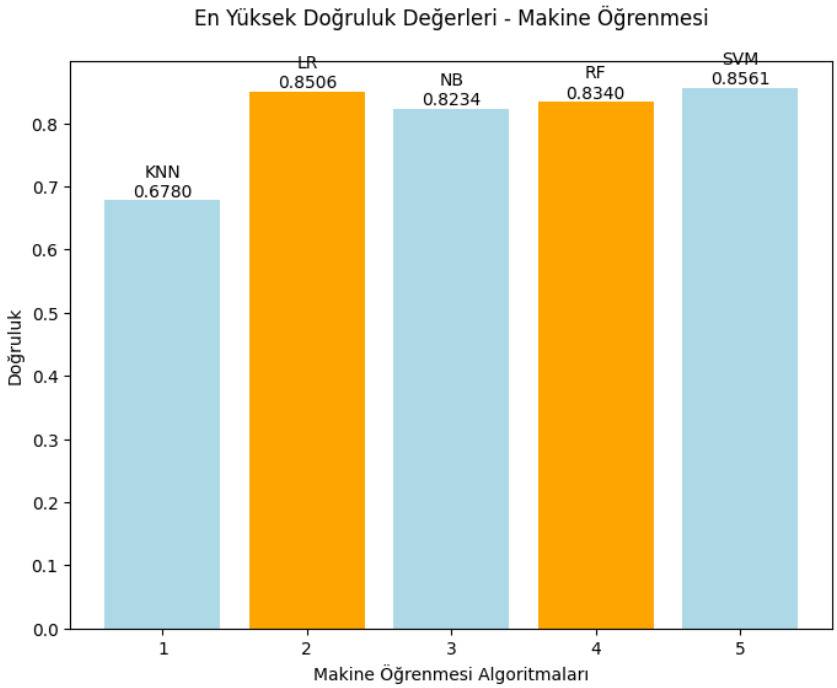


Şekil 3.21 – Rastgele Ormanlar Doğruluk Değerleri Tablosu



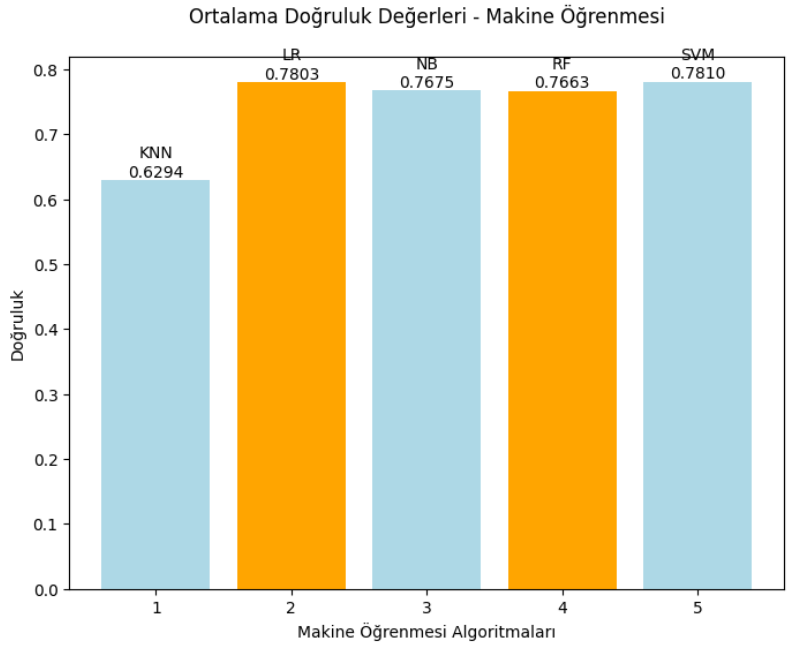
Şekil 3.22 – Destek Vektör Makineleri Doğruluk Değerleri Tablosu

Buna göre makine öğrenmesi algoritmalarının kendi aralarında en yüksek doğruluk değerini SVM algoritması, %85,61'lik oranla elde etmiştir.



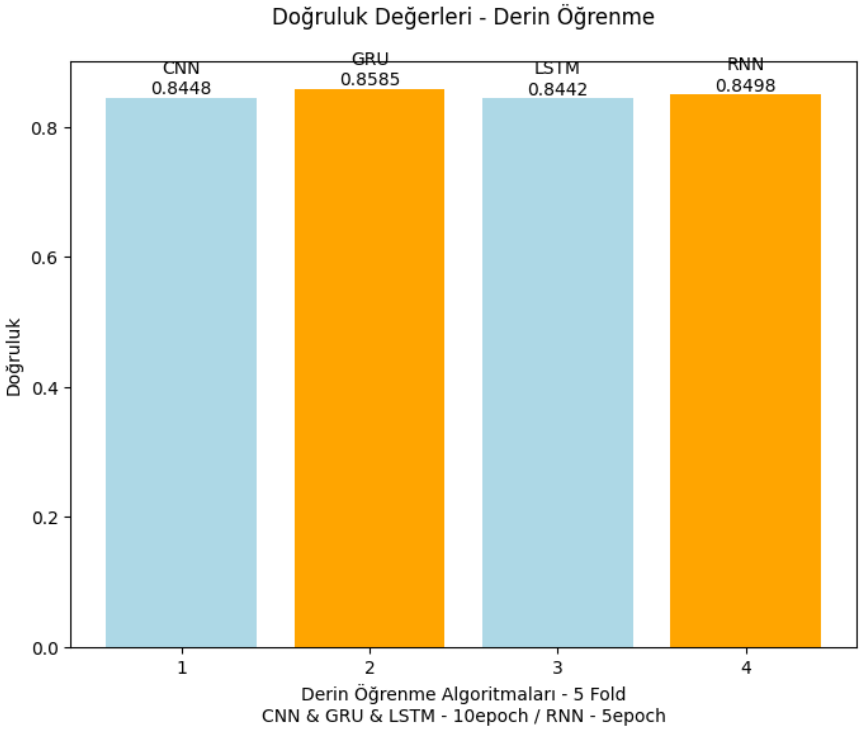
Şekil 3.23 – Makine Öğrenme Algoritmaları En Yüksek Doğruluk Değerleri Tablosu

Ayrıca makine öğrenmesi algoritmalarının farklı n-gram yapılarıyla elde ettikleri doğruluk değerlerinin ortalamaları alındığında; SVM yine %78,1 ile en yüksek değere ulaşmaktadır.



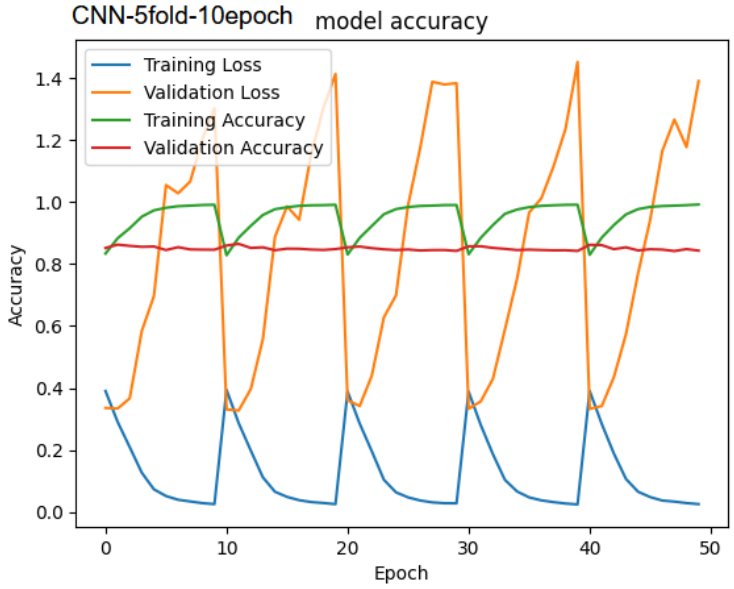
Şekil 3.24 – Makine Öğrenme Algoritmaları Ortalama Doğruluk Değerleri Tablosu

Derin öğrenme algoritmaları arasından ise GRU, %85,85'lik oranla en yüksek doğruluk değerini elde etmiştir.

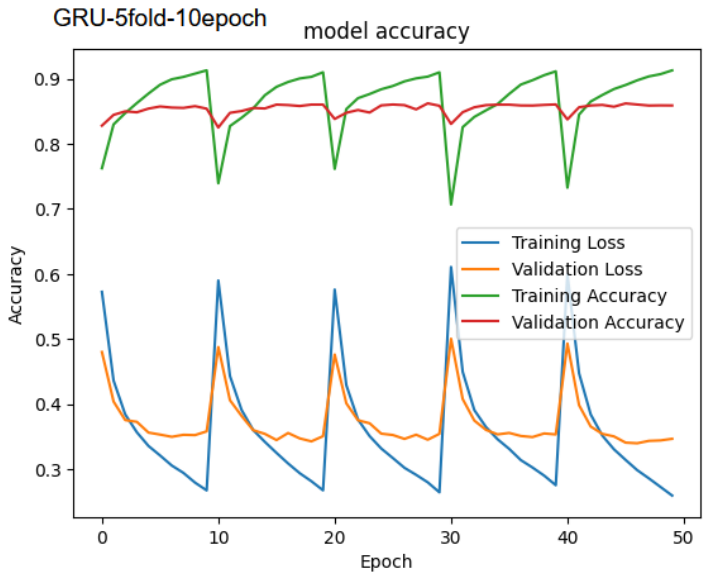


Şekil 3.25 – Derin Öğrenme Algoritmaları En Yüksek Doğruluk Değerleri Tablosu

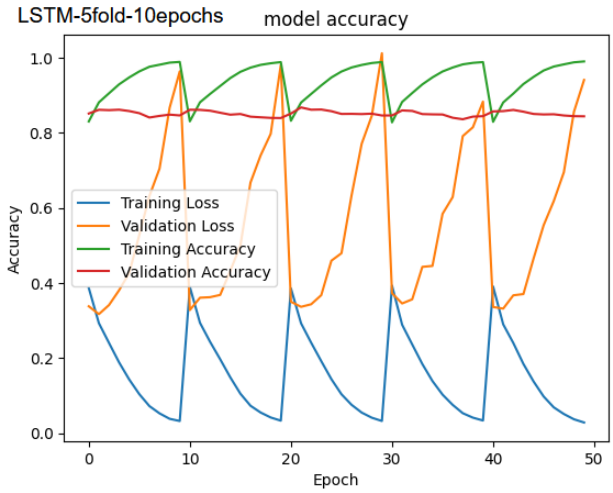
Derin öğrenme algoritmalarının bu değere ulaşırken yürüttüğü epoch süreçlerindeki Doğruluk ve Kayıp Fonksiyonu değerlerinin trafiği aşağıdaki grafiklerle görülebilmektedir:



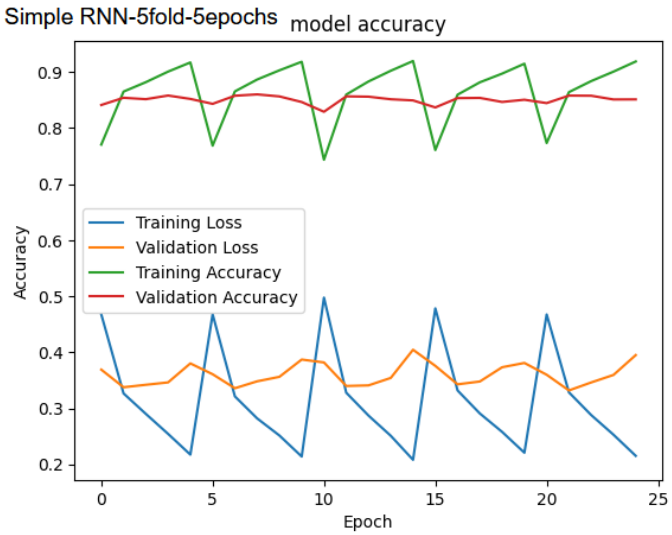
Şekil 3.26 – CNN Algoritması 5 Kat 10 Epoch Süreçleri İzleme Tablosu



Şekil 3.27 – GRU Algoritması 5 Kat 10 Epoch Süreçleri İzleme Tablosu



Şekil 3.28 – LSTM Algoritması 5 Kat 10 Epoch Süreçleri İzleme Tablosu



Şekil 3.29 – Simple RNN Algoritması 5 Kat 5 Epoch Süreçleri İzleme Tablosu

Buna göre derin öğrenme algoritmalarının genel olarak makine öğrenmesi algoritmalarından daha başarılı olduğu söylenebilir. Fakat bu değerlere, algoritmanın düzenlenerek veri kümesini otomatik olarak test ve eğitim kümelerine bölmesi, sonrasında çapraz doğrulama süreçleri ile belirli doğruluk değerlerine ulaşmasıyla erişilmiştir. Burada ilgili modellerin, veri kümesi içindeki değerlere göre ulaştığı değerler söz konusudur. Araştırmacı, kişisel tahminlerine göre bazıları birer, bazıları ise birkaç kelimeden oluşan ifadelerle tüm modellerin ayrıca test edilmesi gerektiğine inanarak; 10 ifadeyle modeli test edebileceğine karar vermiştir.

Araştırmacı; aşağıda yer alan İngilizce 10 ifadenin ilk 6'sıyla önce tüm makine öğrenmesi modellerini, 10'uyla da derin öğrenme modellerini test etmiştir.

İndis	**Test İfadesi**	**İfadenin Türkçe Çevirisi**	**Duygu Durumu (0: Olumsuz, 1: Olumlu)**
1	"good"	İyi	1
2	"bad"	Kötü	0
3	"this is very useful"	Bu çok kullanışlı!	1
4	"it could be better"	Daha iyi olabilirdi...	0
5	"i don't like it at all"	Hiç mi hiç sevmedim!	0
6	"terrible"	Korkunç!	0
7	"awful"	Berbat!	0
8	"perfect"	Mükemmel!	1
9	"does the job"	İş görüyor.	1
10	"doesn't work"	Çalışmıyor.	0

Tablo 3.1 – Manuel Doğruluk Test İfadeleri Tablosu

Buna göre ilgili duygu durumları, 0'lar olumsuz, 1'ler olumlu olacak şekilde sıralı bir diziyle ifade edilebilecektir: (1,0,1,0,0,0,0,1,1,0).

Bu bilgiler ışığında, bu dizi ile; ilgili makine öğrenmesi ve derin öğrenme algoritmalarının verecekleri tepkilere göre oluşturulacak dizilerin, bu gerçek dizi ile ne kadar uyum gösterdiğini bulmak amaçlanmıştır. Diziler arası aynı indisteki elemanların kaç tanesinin birbirine eşit olduğunun yüzdesel oranı, yine başka bir Doğruluk değerini ortaya koymaktadır; zira Doğruluk, gerçekte var olan ile tahmin edilen arası uyumluluğun irdelendiği bir değerdir.

İlgili analiz, Python'daki şu kodlama ile gerçekleştirilmiştir:

```
from sklearn.metrics import classification_report
y_true = [1,0,1,0,0,0,0,1,1,0]
y_pred = [1,0,1,0,1,0,1,1,1,0]
target_names = ['negatif', 'pozitif']
print(classification_report(y_true, y_pred, target_names=target_names))
```

```
              precision    recall  f1-score   support

     negatif       1.00      0.67      0.80         6
     pozitif       0.67      1.00      0.80         4

    accuracy                           0.80        10
   macro avg       0.83      0.83      0.80        10
weighted avg       0.87      0.80      0.80        10
```

Şekil 3.30 – Manuel Ölçev Hesaplaması için Python Kodları Görüntüsü

Kodlamada görüleceği üzere; "y_true" ve "y_pred" adlı iki dizi arası elemanlar karşılaştırılmıştır. Buna göre "y_true" adlı dizi, biraz önce yukarıda verilen test ifadelerinden gelen gerçek duygu durumları; "y_pred" ise kullanılan algoritmalardan birinin gerçekleştirdiği tahminleme ile ortaya çıkan dizidir. Tüm makine öğrenmesi ve derin öğrenme algoritmalarının sonuçları bu yöntemle test edilirken, her seferinde "y_pred" dizisi, çıkan sonuçlara göre elle doldurulmuş; "y_true" dizisi ise gerçeği yansıttığından hep sabit kalmıştır.

Buna göre makine öğrenmesi metotlarının birine uygulanan işlemin sonucunda aşağıdaki görsele ulaşılmıştır:

```
In [14]:
1 newDocument = cv.transform(pd.Series("good"))
2 result = logr.predict(newDocument)
3 result
Out[14]: array([1], dtype=int64)

In [15]:
1 newDocument = cv.transform(pd.Series("bad"))
2 result = logr.predict(newDocument)
3 result
Out[15]: array([0], dtype=int64)

In [16]:
1 newDocument = cv.transform(pd.Series("this is very useful"))
2 result = logr.predict(newDocument)
3 result
Out[16]: array([1], dtype=int64)

In [17]:
1 newDocument = cv.transform(pd.Series("it could be better"))
2 result = logr.predict(newDocument)
3 result
Out[17]: array([0], dtype=int64)

In [18]:
1 newDocument = cv.transform(pd.Series("i don't like it at all"))
2 result = logr.predict(newDocument)
3 result
Out[18]: array([1], dtype=int64)

In [19]:
1 newDocument = cv.transform(pd.Series("terrible"))
2 result = logr.predict(newDocument)
3 result
Out[19]: array([0], dtype=int64)
```

Şekil 3.31 – Manuel Ölçev Hesaplaması Makine Öğrenmesi Tahminlemesi Süreçleri Python Kodları Görüntüsü

Görüldüğü üzere, algoritma; her ifadeyi 0 ya da 1 şeklinde bir sınıfa atamaktadır. İşte çıkan bu değerler, test ifadeleri tablosundaki aynı sırayla alınarak, her seferinde "y_pred" dizisi elle doldurulmuştur.

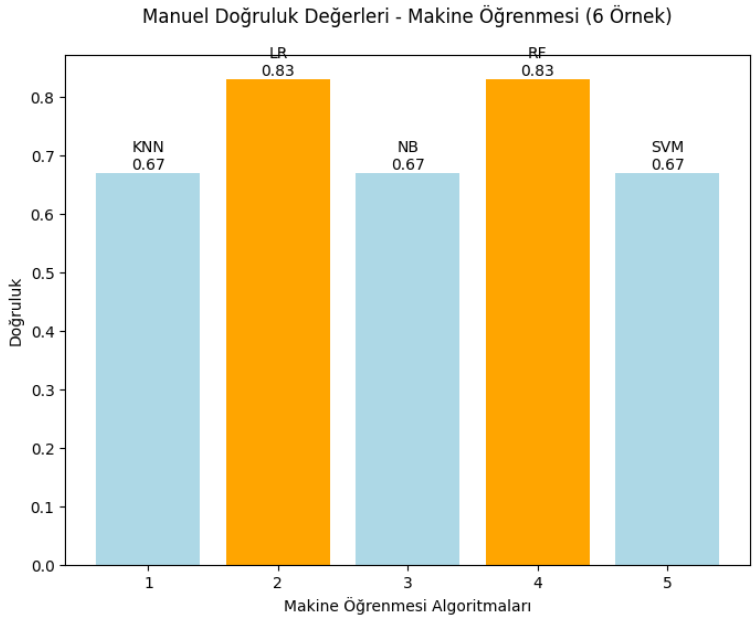
Aşağıda yer alan şekilde de derin öğrenme metotlarından bir tanesinin gerçekleştirdiği sınıflandırma görülebilir:

```
In [26]:
from sklearn.feature_extraction.text import CountVectorizer
# Assuming you have a CountVectorizer object named 'cv'

# Convert your sparse input to dense
newDocument = tokenizer.texts_to_sequences(pd.Series("good"))[0]

# Now you can use this dense input to make predictions with your SVC model
result = model.predict(pad_sequences([newDocument], maxlen=MAX_LENGTH))

result

1/1 [==============================] - 0s 15ms/step

Out[26]: array([[0.08657688, 0.9135572 ]], dtype=float32)

In [27]:
from sklearn.feature_extraction.text import CountVectorizer
# Assuming you have a CountVectorizer object named 'cv'

# Convert your sparse input to dense
newDocument = tokenizer.texts_to_sequences(pd.Series("bad"))[0]

# Now you can use this dense input to make predictions with your SVC model
result = model.predict(pad_sequences([newDocument], maxlen=MAX_LENGTH))

result

1/1 [==============================] - 0s 15ms/step

Out[27]: array([[0.8651208 , 0.13477127]], dtype=float32)
```

Şekil 3.32 – Manuel Ölçev Hesaplaması Derin Öğrenme Tahminlemesi Süreçleri Python Kodları Görüntüsü

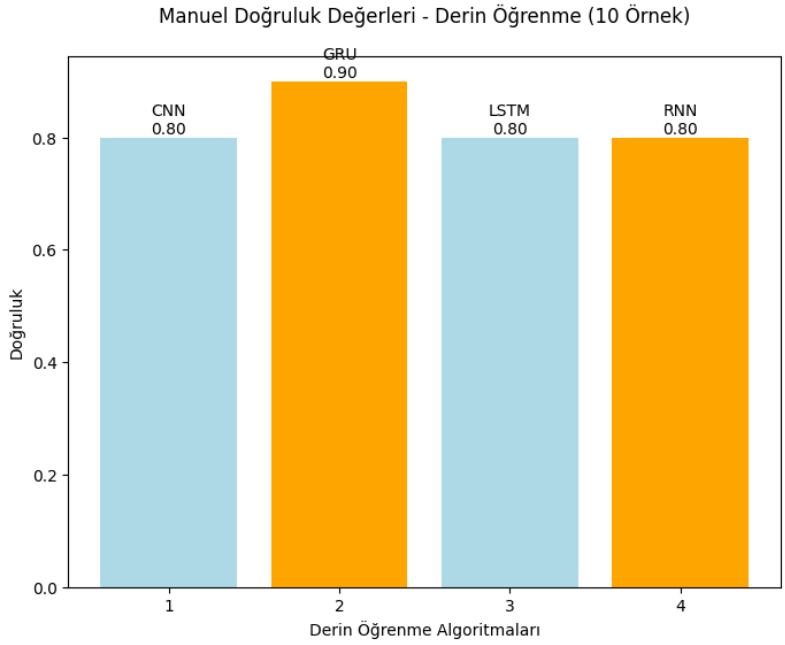
Görüldüğü gibi, sol taraf 0 (olumsuz), sağ taraf 1 (olumlu) olarak düşünüldüğünde; söz konusu algoritma, “good” ifadesinin %91 olasılıkla olumlu, “bad” ifadesinin de %86,5 olumsuz olduğu sonucuna varmıştır.

Bu şekildeki makine öğrenmesi doğruluk belirleme işlemleri sonucunda Lojistik Regresyon ve Rastgele Ormanlar algoritmaları %83’lük başarı göstermişlerdir.



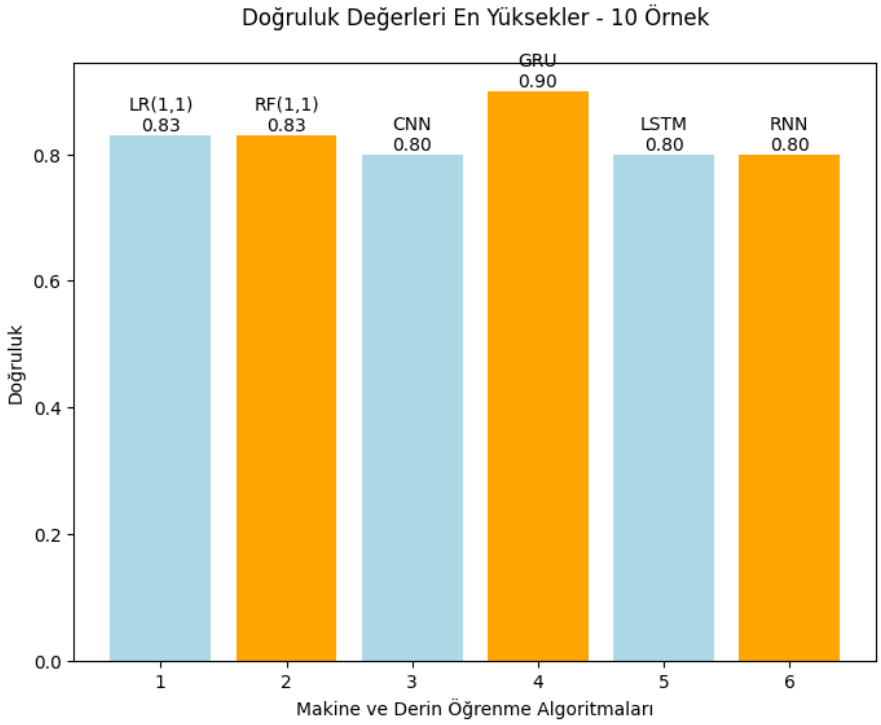
Şekil 3.33 – Makine Öğrenmesi 6 Örnekle Manuel Hesaplama Doğruluk Değerleri

Derin öğrenme algoritmaları arasından ise GRU, %90'lık oran ile en yüksek doğruluk tahmini başarısını; diğer derin öğrenme algoritmaları ise %80 başarı göstermişlerdir.



Şekil 3.34 – Derin Öğrenme 10 Örnekle Manuel Hesaplama Doğruluk Değerleri

Bu işlemlerden sonra yapılması gereken tek şey; en başarılı makine öğrenmesi algoritmaları olan Lojistik Regresyon ve Rastgele Ormanlar algoritmalarını da 10 örnekli teste tabi tutmak olmuştur. Bu sefer 10 ifadenin tamamı, bu iki algoritmanın uni-gram yapılı kodlarıyla test edilmiştir. Bu uygulamayla iki algoritma da %80'lik başarı göstermiş; GRU dışındaki derin öğrenme algoritmalarıyla aynı seviyeye yükselmiştir. Fakat diğer hiçbir algoritma gibi bu iki algoritma da %90 doğruluk değerine ulaşan GRU algoritmasını geçememiştir.



Şekil 3.35 – 10 Örnekle Manuel Hesaplama En Yüksek Doğruluk Değerli Tüm Algoritmalar

Bölüm 4

Sonuç

Çalışmamızda Amazon.com sitesinden web kazıma yöntemiyle elde edilen 50.000 satırlık bir veri kümesi üzerinde gerekli ön işlemler gerçekleştirildikten sonra 5 makine öğrenmesi ve 4 derin öğrenme algoritması uygulanmış; alınan en yüksek doğruluk değerinin **%85,85'lik bir oran ile GRU (Gated Recurrent Units - Geçitli Tekrarlayan Birimler)** algoritması tarafından elde edildiği görülmüştür. Buna ek olarak veri kümesi dışından, araştırmacının bireysel tahminleriyle belirlediği 10 ifade ile tüm algoritmalar tekrar teste tabi tutulmuş; ilk olarak 6 ifade ile test edilen makine öğrenmeleri arasından **Lojistik Regresyon ve Rastgele Ormanlar algoritmalarının %83'lük doğruluk oranına** ulaştıkları görülmüştür. Bu algoritmalar 10 ifade ile tekrar test edilerek **%80'lik doğruluk oranına** ulaşmışlardır. %80 doğruluk oranı elde edilen 3 Derin Öğrenme Algoritması ve 2 Makine Öğrenmesi algoritması; **%90'lık doğruluk oranı** elde eden **GRU (Gated Recurrent Units - Geçitli Tekrarlayan Birimler)** algoritmasını geçememiştir. **Buna göre GRU, tüm algoritmalar arasında en başarılı sonuçlara ulaşan algoritma olmuştur.**

Bu çalışma ile derin öğrenme algoritmalarının, makine öğrenmesi algoritmalarından, genel anlamda, daha başarılı bir şekilde duygu analizi yapabildiği sonucuna ulaşılmıştır. Derin öğrenme algoritmaları arasında da GRU'nun en yüksek başarıya ulaştığı görülmüştür.

Ancak test ifadeleri kümesi, 10 ifadeden oluşmaktadır ve bu küme geliştirilebilir. Gelecekteki bir çalışmada 50.000 yerine, çok daha fazla büyüklükteki ve farklı ifadelere sahip veri kümelerinin elde edilmesi planlanabilir. Bu veri kümesinin de 10 yerine, örneğin 100 ifadeyle test edilmesi sağlanabilir. Fakat bu tarz bir çalışma gerçekleştirilirken, homojen ifadelere sahip olması beklenen benzer ürünlerle mi yoksa birbirinden farklı kategorilerden ürünlerle mi çalışılması gerektiğine karar verilmesi gerekecektir.

Kaynakça

AALAMİ, N. (2020), "DERİN ÖĞRENME YÖNTEMLERİNİ KULLANARAK GÖRÜNTÜLERİN ANALİZİ", ESTUDAM Bilişim Dergisi, 1(1),17–20.

ABIODUN, O. I., JANTAN, A., OMOLARA, A. E., DADA, K. V., MOHAMED, N. A., & ARSHAD, H. (2018). STATE-OF-THE-ART IN ARTIFICIAL NEURAL NETWORK APPLICATIONS: A SURVEY. Heliyon Publishing, 4(11).

AFŞAR, M. (2001). E-TİCARET VE BANKALARIN ROLÜ. Anadolu Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi, 17(1), 189-229.

ALAGHA, B. (2023); XSS ATTACK DETECTION WITH N-GRAM BASED PREDICTION MODEL", ESTUDAM Bilişim Dergisi, 4(2), 1–9.

AYDIN ATASOY, N., & TABAK, D. (2018). DESTEK VEKTÖR MAKİNELERİ KULLANARAK YÜZ TANIMA UYGULAMASI GELİŞTİRİLMESİ. Engineering Sciences, 13(2), 119-127.

AYTEKİN, Ç., & BAYRAM, M. A. (2021). TÜRKÇE METİNLER İÇİN DUYGU ANALİZİ YAKLAŞIMI İLE İLETİŞİMDE BAĞLAMDAN BAĞIMSIZ MODELLERİN GELİŞTİRİLMESİ ÜZERİNE BİR ARAŞTIRMA: KARMA VERİ MODELİ ÖNERİSİ. Yeni Medya Elektronik Dergisi, 5(1), 12-25.

BARUT, Z., & ALTUNTAŞ, V. (2023). COMPARISON OF PERFORMANCE OF DIFFERENT K VALUES WITH K-FOLD CROSS VALIDATION IN A GRAPH-BASED LEARNING MODEL FOR IncRNA-DISEASE PREDICTION. Kırklareli Üniversitesi Mühendislik Ve Fen Bilimleri Dergisi, 9(1), 63-82.

BAYAT, S., & IŞIK, G. (2023). EVALUATING THE EFFECTIVENESS OF DIFFERENT MACHINE LEARNING APPROACHES FOR SENTIMENT CLASSIFICATION. Journal of the Institute of Science and Technology, 13(3), 1496-1510.

BAYRAK, T. (2023). E-TİCARETTE MÜŞTERİ MAĞDURİYETİ: TRENDYOL ÖRNEĞİ. Yeni Yüzyıl'da İletişim Çalışmaları, 2(6), 67-80.

BİLGİN, M., & ŞENTÜRK, İ. F. (2019). DANIŞMANLI VE YARI DANIŞMANLI ÖĞRENME KULLANARAK DOKÜMAN VEKTÖRLERİ TABANLI TWEETLERİN DUYGU ANALİZİ. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 21(2), 822-839.

BOSTANCI, B., & ALBAYRAK, A. (2021). DUYGU ANALİZİ İLE KİŞİYE ÖZEL İÇERİK ÖNERMEK. Veri Bilimi Dergisi, 4(1), 53-60.

CAVNAR, W. B., TRENKLE, J. M. 1994. N-GRAM-BASED TEXT CATEGORIZATION. In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval (Vol. 161175)

ÇINAROĞLU, E., & AVCI, T. (2020). THY Hisse Senedi Değerinin Yapay Sinir Ağları İle Tahmini. Atatürk Üniversitesi İktisadi Ve İdari Bilimler Dergisi, 34(1), 1-19.

CORTES, C & VAPNIK, V. (1995) "SUPPORT-VECTOR NETWORKS", Machine Learning, 20, 273-297, Kluwer Academic Publishers.

Crummy.com; “BEAUTIFUL SOUP DOCUMENTATION”; https://www.crummy.com/software/BeautifulSoup/bs4/doc/ ; Erişim: 26.01.2024

DİNÇER, E. Ş.; KAYAOĞLU, D & SAFARLI, S. (2022), S, METİN MADENCİLİĞİ VE DUYGU ANALİZİ İLE SİBER ZORBALIK TESPİTİ”, ESTUDAM Bilişim Dergisi, 3(2), 38–45.

DU, Ke-Lin & SWAMY, M.N.s. (2014). NEURAL NETWORKS AND STATISTICAL LEARNING. Pg.18. Springer Publishing.

GÖÇGÜN, Ö. F., & ONAN, A. (2021). AMAZON ÜRÜN DEĞERLENDİRMELERİ ÜZERİNDE DERİN ÖĞRENME/MAKİNE ÖĞRENMESİ TABANLI DUYGU ANALİZİ YAPILMASI. Avrupa Bilim Ve Teknoloji Dergisi (24), 445-448.

HARK, C., KARCI, A., SEYYARER, E. & UÇKAN, T. (2019) AĞIRLIKLANDIRILMIŞ ÇİZGELERDE TF-IDF VE EIGEN AYRIŞIMI KULLANARAK METİN SINIFLANDIRMA”, Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 8(4), 1349–1362.

Ibm.com; “WHAT ARE RECURRENT NEURAL NETWORKS?” https://www.ibm.com/topics/recurrent-neural-networks; Erişim: 30.01.2024

Ibm.com; “STRUCTURED VS. UNSTRUCTURED DATA: WHAT’S THE DIFFERENCE?” https://www.ibm.com/blog/structured-vs-unstructured-data/; Erişim: 27.01.2024

İLHAN, N., & SAĞALTICI, D. (2020). TWITTER’DA DUYGU ANALİZİ. Harran Üniversitesi Mühendislik Dergisi, 5(2), 146-156.

KUMOVA METİN, S. & KARAOĞLAN, B. (2017). STOP WORD DETECTION AS A BINARY CLASSIFICATION PROBLEM. Anadolu University Journal of Science and Technology A - Applied Sciences and Engineering, 18(2), 346-359.

NACAR, E. N., & ERDEBİLLİ (B.D.ROUYENDEGH), B. (2021). MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE SATIŞ TAHMİNİ. Endüstri Mühendisliği Dergisi, 32(2), 307-320.

ORHAN, U., ADEM, K., & COMERT, O. (2012). LEAST SQUARES APPROACH TO LOCALLY WEIGHTED NAIVE BAYES METHOD. Journal of New Results in Science, 1(1), 71-80.

PARLAK, M.S., & KAYRİ, M. (2022). ÖĞRETMENLERİN E-ÖĞRENME HAZIRBULUNUŞLUK DÜZEYLERİNİ ETKİLEYEN FAKTÖRLERİN RASTGELE ORMAN ALGORİTMASI YÖNTEMİ İLE İNCELENMESİ. Van Yüzüncü Yıl Üniversitesi Eğitim Fakültesi Dergisi, 19(3), 670-696.

POONGODAI, A. & SUHASINI, R.(2019). A COMMAND LINE TOOL FOR TRACKING ERROR DETAILS OF PROGRAM USING WEB SCRAPER. International Journal of Recent Technology and Engineering (IJRTE), 8(2S11), 2277-3878.

Pypi.org; “REQUESTS”; https://pypi.org/project/requests/; Erişim: 27.01.2024

Python.com; “WHAT IS PYTHON? EXECUTIVE SUMMARY”; https://www.python.org/doc/essays/blurb/ ; Erişim: 27.01.2024

SABANCI, K. (2016). DIFFERENT APPLE VARIETIES CLASSIFICATION USING KNN AND MLP ALGORITHMS. International Journal of Intelligent Systems and Applications in Engineering, 4(Special Issue-1), 166-169.

ŞAHİNASLAN, Ö., DALYAN, H., & ŞAHİNASLAN, E. (2022). Naive Bayes Sınıflandırıcısı Kullanılarak YouTube Verileri Üzerinden Çok Dilli Duygu Analizi. Bilişim Teknolojileri Dergisi, 15(2), 221-229.

ŞENEL, S., & ALATLI, B. (2014). LOJİSTİK REGRESYON ANALİZİNİN KULLANILDIĞI MAKALELER ÜZERİNE BİR İNCELEME. Journal of Measurement and Evaluation in Education and Psychology, 5(1), 35-52.

Simplilearn.com; "WHAT IS EPOCH IN MACHINE LEARNING?" https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning; Erişim: 31.01.2024

Sozluk.gov.tr; "YORDAMAK"; https://www.sozluk.gov.tr/; Erişim: 29.01.2024

Splash.readthedocs.io; SPLASH – "A JAVASCRIPT RENDERING SERVICE"; https://splash.readthedocs.io/en/stable ; Erişim: 26.01.2024

TAHİROĞLU, B. T. (2021). LEMATİZASYON VE TÜRKÇE İÇİN BİR LEMATİZASYON UYGULAMASI: ELEMANTR. RumeliDE Dil Ve Edebiyat Araştırmaları Dergisi (24), 475-486.

TAYE, Mohammad. (2023). THEORETICAL UNDERSTANDING OF CONVOLUTIONAL NEURAL NETWORK: CONCEPTS, ARCHITECTURES, APPLICATIONS, FUTURE DIRECTIONS. Computation, MDPI Publishing. 11(52), 2,12.

Wikimedia.org; "GATED RECURRENT UNIT"; https://upload.wikimedia.org/wikipedia/commons/3/37/Gated_Recurrent_Unit%2C_base_type.svg; Erişim: 30.01.2024

Wikipedia.org; "HTTP"; https://en.wikipedia.org/wiki/HTTP; Erişim: 31.01.2024

Wikimedia.org.; “LSTM CELL”; https://upload.wikimedia.org/wikipedia/commons/9/93/LSTM_Cell.svg; Erişim: 30.01.2024

Wikimedia.org; “RECURRENT NEURAL NETWORK UNFOLD”; https://upload.wikimedia.org/wikipedia/commons/b/b5/Recurrent_neural_network_unfold.svg; Erişim: 30.01.2024

Wikipedia.org; “AMAZON (COMPANY)”; https://en.wikipedia.org/wiki/Amazon_(company); Erişim: 09.06.2023

Wikipedia.org; "CONVOLUTION"; https://en.wikipedia.org/wiki/Convolution; Erişim: 30.01.2024

Wikipedia.org; “GATED RECURRENT UNIT”; https://en.wikipedia.org/wiki/Gated_recurrent_unit; Erişim: 30.01.2024

Wikipedia.org; "LANGUAGES OF THE UNITED STATES"; https://en.wikipedia.org/wiki/Languages_of_the_United_States; Erişim: 27.01.2024

Wikipedia.org; “LOGISTIC REGRESSION”; https://en.wikipedia.org/wiki/Logistic_regression; Erişim 29.01.2024

Wikipedia.org; “PRECISION AND RECALL”; https://en.wikipedia.org/wiki/Precision_and_recall, Erişim: 30.01.2024

Wikipedia.org; “STOCHASTIC PROCESS”; https://en.wikipedia.org/wiki/Stochastic_process; Erişim: 29.01.2024

Wikipedia.org; “YOUTUBER”; https://en.wikipedia.org/wiki/YouTuber ; Erişim: 26.01.2024

YouTube; ROONEY, J.W. (2020). HOW I SCRAPE AMAZON REVIEWS USING PYTHON, REQUESTS & BEAUTIFULSOUP; https://www.youtube.com/watch?v=DIT8rwyPEns; Erişim: 27.01.2024

YURTSEVER, M. (2021). GOLD PRICE FORECASTING USING LSTM, BI-LSTM AND GRU. Avrupa Bilim Ve Teknoloji Dergisi(31), 341-347.